



Universidad
Carlos III de Madrid

Departamento de Ingeniería Telemática

TRABAJO FIN DE GRADO

Integración de sensor de movimiento en Unity- 3D para monitorizar la movilidad de usuarios en silla de ruedas

Autor: José Antonio Díaz Jaime

Tutor: M^a del Carmen Fernández Panadero

Leganés, Junio 2014

Título: Integración de sensor de movimiento en Unity-3D para monitorizar la movilidad de usuarios en silla de ruedas.

Autor: José Antonio Díaz Jaime

Tutor: M^a del Carmen Fernández Panadero

EL TRIBUNAL

Presidente: Pablo Acedo Gallardo

Vocal: Norberto Fernández García

Secretario: Alejandro García Lampérez

Realizado el acto de defensa y lectura del Trabajo de Fin de grado el día 15 de Julio de 2014, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Para comenzar la memoria, me gustaría acordarme de una serie de personas que han sido una fuente de inspiración y un sólido sostén en el cual poder ayudarme en momentos de dificultad.

En primer lugar, acordarme de mis padres, José Luis y Ana, los cuales desde pequeño me han inculcado el ser responsable, trabajar cada día más que el anterior y nunca perder el interés por seguir aprendiendo. Me han ofrecido dos visiones de la vida muy diferentes y muy compatibles a la vez y la verdad es que siempre han tenido palabras de ánimo y predisposición total para cualquier cosa que he necesitado.

Destacar a mi hermano Alberto, ese hermano mayor que siempre busca lo mejor para ti, te anima, en el que puedes apoyarte cuando lo necesitas y el que comparte todos los años de tu vida.

Acordarme también de amigos y compañeros de carrera, que han hecho que estos cuatro años fueran parte importante de lo que soy hoy en día y la base de lo que seré el día de mañana.

Agradecer de forma notoria a mi tutora, M^a Carmen Fernández Panadero el permitirme trabajar con ella y el estar disponible para cualquier ayuda que necesitara. La dedicación incondicional a su trabajo deja huella en sus alumnos y es sin lugar a dudas uno de los mejores profesores que he tenido en todos mis años de estudiante, un verdadero ejemplo en el que mirarse.

Gracias por supuesto a la empresa Simumak, que ha permitido que realice prácticas en su empresa y de la que he obtenido una primera

experiencia en el mundo laboral muy enriquecedora, y también a Valentín, autor del desarrollo base del que he partido y que siempre ha estado disponible para solucionar cualquier duda que pudiera tener con su trabajo previo.

Para terminar, dar las gracias a todo el personal docente por el trabajo y su dedicación a lo largo de toda la carrera.

Resumen

Tecnologías como la simulación, permiten trabajar en un entorno seguro que proteja en todo momento al usuario. El principal auge se está viviendo en el campo de la medicina-terapia, donde la posibilidad de llevar un registro del paciente en tiempo real o realizar pruebas con un coste personal y económico bajo, suponen una gran alternativa a los métodos tradicionales.

En esta vía de trabajo se ha decidido aplicar tecnologías de simulación para monitorizar el movimiento de un usuario discapacitado. El desarrollo ha comprendido tareas de integración hardware (en la que destaca un dispositivo de detección de movimiento) y software, utilizando el motor de videojuegos de Unity3D como herramienta principal.

Se ha conseguido obtener un simulador que permite al usuario visualizar su movimiento en un entorno virtual a la vez que realiza ejercicios interactivos y guiados en un escenario acorde a la temática de los mismos. De esta forma, el proyecto supone una solución efectiva tanto para el aprendizaje como para la familiarización con actividades comunes en los discapacitados.

Por último, este simulador permitirá experimentar en primera persona las dificultades con las que se encuentra una persona discapacitada a la hora de realizar tareas tan simples para una persona con movilidad total.

Palabras clave: Simulador, discapacitado, monitorización, aprendizaje, medicina.

Abstract

Technologies such as simulation, allow working in a secure environment that protects the user all time. Simulation has mainly risen in popularity in the environment of medical-therapy, where the opportunity to keep real-time tracking of patients or testing with low personal and economic costs, represent a great alternative to traditional methods.

In this project, it has been decided to apply simulation technologies to monitor the movement of a disabled user so that the feeling of immersion is as close as possible to reality. Development includes hardware integration tasks (which highlights a motion detection device) and software, using the Unity3D game engine as the main tool.

It has been possible to obtain a simulator that allows users to visualize their movement in a virtual environment while doing interactive and guided exercises in a scene according to the theme of the simulation. In this way, the project has become an effective solution for learning and familiarization with common activities in the disabled people.

Finally, this simulator allows the users to experience for themselves the difficulties that disabled people face when performing tasks that are simple for a person with full mobility.

Keywords: Simulator, disabled, monitoring, learning, medicine, therapy.

Índice general

1. Introducción y Objetivos	1
1.1. Introducción	1
1.2. Entorno socioeconómico y marco regulador	3
1.3. Motivación del proyecto	5
1.4. Objetivos	6
1.5. Estructura de la memoria	11
2. Estado del Arte	13
2.1. Introducción	13
2.2. Análisis de los simuladores de apoyo a discapacitados existentes	15
2.3. Conclusiones	21
2.4. Tecnologías empleadas	25
2.4.1. Unity3D PRO v.4.3.4f1	25
2.4.2. SketchUp.....	28
2.4.3. Blender.....	30

2.4.4. ASUS XtionPRO Live	33
3. Diseño del Sistema.....	37
3.1. Introducción	37
3.2. Diseño de ejercicios	41
3.3. Modelado y Desarrollo en Unity3D.....	44
3.3.1. Container.....	45
3.3.2. Escenarios y componentes	46
3.3.3. Personajes	52
3.3.4. Puntos de control y audio	55
3.3.5. Gestión de cámaras y navegación.....	56
3.4. Estructura del simulador	58
3.4.1. Comunicación hardware	58
3.4.2. Diseño y estructura software	60
3.5. Integración hardware.....	68
3.5.1. Dispositivo de detección de movimiento.....	68
3.5.2. Dispositivos HMD	69
3.5.3. Logitech Dual Action	72
3.6. Requisitos del simulador.....	73
3.6.1. Requisitos Funcionales	74
3.6.2. Requisitos No Funcionales	78
4. Plan de validación	81

4.1. Introducción	81
4.2. Pruebas realizadas	82
4.3. Recursos utilizados para las pruebas	83
4.4. Requisitos del sistema	84
5. Planificación del proyecto	85
5.1. Ciclo de vida	85
5.2. Organización	87
5.3. Plan de trabajo.....	90
5.4. Presupuesto	91
5.4.1. Coste de personal	91
5.4.2. Coste del equipo	92
5.4.3. Coste de licencias	93
5.4.4. Resumen de costes	94
6. Resultados y Conclusiones.....	97
7. Futuras Líneas de Trabajo.....	101

Índice de figuras

Figura 1: Ejercicios estandarizados para personas discapacitadas	4
Figura 2: Oculus Rift y Vuzix	9
Figura 3: Simulador Toyra.....	17
Figura 4: Simulador DriveSafety CDS-250W desarrollado por la Universidad Clemson	18
Figura 5: Simulador para pacientes con ictus desarrollado por CIDIF	19
Figura 6: Simulador desarrollado en la PUCP	21
Figura 7: Logo de Unity	25
Figura 8: Entorno de trabajo de Unity3D	27
Figura 9: Logo SketchUp.....	28
Figura 10: Entorno de trabajo de SketchUp	29
Figura 11: Logo Blender.....	30
Figura 12: Entorno de trabajo de Blender.....	32
Figura 13: Dispositivo ASUS XtionPRO LIVE	33
Figura 14: Detección de movimiento con XtionPRO LIVE.....	34

Figura 15: Pantalla de acceso al sistema.....	38
Figura 16: Ejercicio 1 – Transición silla de ruedas/cama.....	39
Figura 17: Ejercicio 2 – Recoger objetos de una cocina	39
Figura 18: Ejercicio 3 – Fortalecimiento de brazos.....	40
Figura 19: Visión escena en primera persona vs tercera persona	41
Figura 20: Desglose del primer ejercicio de entrenamientos.....	42
Figura 21: Desglose del segundo ejercicio de entrenamientos	43
Figura 22: Desglose del tercer ejercicio de entrenamientos	44
Figura 23: Modelo inicial del exterior del container de entrenamientos	45
Figura 24: Modelo final del container de entrenamientos	46
Figura 25: Modelo inicial del ejercicio 1 (Habitación).....	47
Figura 26: Modelo final del ejercicio 1 (Habitación)	48
Figura 27: Modelo inicial del ejercicio 2 (Cocina).....	49
Figura 28: Modelo final del ejercicio 2 (Cocina)	50
Figura 29: Modelo inicial del ejercicio 3 (Salón).....	51
Figura 30: Modelo final del ejercicio 3 (Salón).....	52
Figura 31: Selección de género de personaje	53
Figura 32: Modelo inicial hombre vs modelo final hombre	54
Figura 33: Modelo inicial mujer vs modelo final mujer.....	55
Figura 34: Transición entre puntos de control (ej: ejercicio 3)	56
Figura 35: Vista primera persona vs vista tercera persona	57

Figura 36: Vista girada a la izquierda vs vista girada a la derecha.....	57
Figura 37: Esquema de comunicaciones a nivel hardware	59
Figura 38: Estructura Software en directorios	60
Figura 39: Desglose del directorio “Scripts”	62
Figura 40: Esquema de los sistemas que componen el Software del simulador	63
Figura 41: Gestión de escuchadores de Zig.cs.....	64
Figura 42: Engaged de usuario	65
Figura 43: Modificación de las características de la silla de ruedas	66
Figura 44: Asus XtionPRO LIVE integrado en el proyecto	68
Figura 45: Vuzix iWear VR920 integrada en el proyecto	70
Figura 46: Oculus Rift VR integrada en el proyecto	71
Figura 47: Logitech Dual Action integrado en el proyecto	72
Figura 48: Detección de movimiento vertical de Joystick.....	73
Figura 49: Detección de pulsación de botón1	73
Figura 50: Ciclo de vida	87
Figura 51: Diagrama de Gantt del proyecto	89
Figura 52: Selección de modo de juego (Menú Inicial)	110
Figura 53: Selección de género (Menú Inicial)	110
Figura 54: Selección del momento del día (Menú Inicial)	111
Figura 55: Menú Pausa	111
Figura 56: Registro de logros (Menú Pausa)	112

Figura 57: Selección de ejercicio.....	113
Figura 58: Ejercicio confirmado y comienza la monitorización	113
Figura 59: Cambio de cámara (Modalidade entrenamientos).....	114
Figura 60: Giro de cámara (Modalidade entrenamientos).....	114
Figura 61: Ejercicio entrenamiento pausado	115
Figura 62: Controles del Joystick	116
Figura 63: Conexiones de red	117
Figura 64: Propiedades de Conexión de área local.....	118
Figura 65: Propiedades de protocolo (TCP / IPV4).....	118
Figura 66: Montaje gorra y reflector (Track Clip).....	120
Figura 67: Constantes de control (AdminEscena.cs).....	121
Figura 68: Instancia de personajes (AdminEscena.cs)	122
Figura 69: Update AdminEscena.cs.....	123
Figura 70: Constantes de control (ManageExercises.cs).....	123
Figura 71: Selección de ejercicio (ManageExercises.cs).....	124
Figura 72: Update ManageExercises.cs.....	125
Figura 73: Método OnTriggerEnter (Ejercicios monitorizados)	125
Figura 74: Gestión de jerarquías y colores (Ejercicios monitorizados).....	126
Figura 75: Constantes de control (Ejercicios monitorizados).....	126

Índice de tablas

Tabla 1: Resumen de los proyectos científicos / simuladores estudiados anteriormente	23
Tabla 2: Resumen de las principales características de las tecnologías empleadas durante el desarrollo	36
Tabla 3: Requisito funcional RF – 01	74
Tabla 4: Requisito funcional RF – 02.....	74
Tabla 5: Requisito funcional RF – 03.....	75
Tabla 6: Requisito funcional RF – 04.....	75
Tabla 7: Requisito funcional RF – 05.....	76
Tabla 8: Requisito funcional RF – 06.....	76
Tabla 9: Requisito funcional RF – 07.....	76
Tabla 10: Requisito funcional RF – 08.....	77
Tabla 11: Requisito funcional RF – 09.....	77
Tabla 12: Requisito funcional RF – 10.....	78
Tabla 13: Requisito no funcional RNF – 01	78
Tabla 14: Requisito no funcional RNF – 02	79

Tabla 15: Matriz Objetivos vs Requisitos	80
Tabla 16: Características del equipo de testeo	83
Tabla 17: Coste de personal.....	92
Tabla 18: Coste del equipo	93
Tabla 19: Coste de licencias	94
Tabla 20: Coste general	94

Capítulo 1

Introducción y Objetivos

En este apartado se va a analizar la situación en la que se realiza el proyecto, cuales son los principales motivos para su desarrollo y qué objetivos se han planteado conseguir.

1.1. Introducción

La era digital y de desarrollo tecnológico sin límites en la que se está viviendo actualmente, provoca el surgimiento de nuevos dispositivos que satisfagan necesidades que antes estaban por resolver como puede ser el disponer de conexión a internet en el móvil, controlar electrodomésticos de forma remota, pantallas con visión en 3D, etc.

En este entorno es donde aparece una de las industrias que están viviendo una clara época de auge tanto de desarrollo como comercial, los simuladores. En España hay grandes empresas como Indra¹, tradicionalmente dedicadas al mundo de la simulación, pero en las últimas décadas han surgido otras empresas más pequeñas como Simumak (Simumak²) y Future Platone (FP³) que están acercando el mundo de los simuladores a clientes con menor

¹ <http://www.indracompany.com/>

² <http://simumak.com>

³ <http://www.futureplatone.com>

capacidad comercial, contribuyendo de este modo a la expansión de esta tecnología.

Esta herramienta proporciona muchas posibilidades en el mundo del aprendizaje porque permite involucrar al alumno desde el punto de vista físico, mental y emocional. Prueba de ello es la enorme cantidad de aplicaciones de esta tecnología en diferentes áreas como la medicina, el deporte, la terapia y el ocio.

Gran parte del potencial de esta tecnología es que permite recrear situaciones reales en entornos virtuales que proporcionen seguridad al usuario que los maneja y que permita a través de la práctica, mejorar sus habilidades, con el fin de aplicarlas posteriormente a su vida cotidiana. Esta posibilidad de fallar de forma segura hace de los simuladores una herramienta ideal para el aprendizaje por competencias. Aunque la mayoría de los casos requiere de una experiencia real para concluir el aprendizaje, es notoria la utilidad y las ventajas que proporciona este tipo de herramientas como toma de contacto inicial para acelerar el aprendizaje en casos en los que la práctica en entornos reales no sea posible.

Una de las principales características de los simuladores y que supone un aspecto clave a la hora de satisfacer la necesidad del usuario, es la de conseguir el mayor nivel de realismo posible, permitiendo al usuario abstraerse de que está utilizando una herramienta virtual de forma que la experiencia sea lo más enriquecedora posible. Autores como Choi [1] han demostrado el impacto de la interactividad y el realismo en la satisfacción del estudiante y por tanto en los resultados del aprendizaje.

Sin embargo, dejando a un lado los beneficios de la interactividad, el hacer uso de simuladores permite al usuario una formación en competencias, en el caso del simulador que nos ocupa, competencias instrumentales. Estas competencias tienen una función instrumental y pueden ser metodológicas, físicas o cognoscitivas.

El simulador que se desarrolla en este proyecto permite al usuario adquirir una serie de competencias tales como:

- Enfrentarse a problemas más complejos para contribuir a la capacidad de tomar decisiones de manera razonada y resolver problemas.
- Destrezas físicas relacionadas con la utilización del propio simulador.
- Entrenamiento en tareas sencillas que permitan capacidad de análisis y asimilación de las metodologías empleadas en cada ejercicio.

Como se puede dilucidar, la herramienta que se va a diseñar va a permitir al usuario discapacitado facilitar la identificación de las habilidades necesarias para desempeñar con éxito en el mundo real tareas simuladas y al usuario sin discapacidades concienciarse de las dificultades que presentan ejercicios aparentemente sencillos.

1.2. Entorno socioeconómico y marco regulador

El aumento progresivo del número de discapacitados, que según estudios de la Organización Mundial de la Salud representan el 10% de la población mundial [24], está agudizando cada vez más el problema de hacer accesibles las ciudades a las personas con discapacidad [25]. Con el horizonte 2020 [26], la Comisión Europea identifica varios retos prioritarios relacionados con estos temas (salud, bienestar y ciudades integradoras). En este sentido, se han desarrollado en las últimas décadas múltiples simuladores orientados a personas en silla de ruedas [27], habiéndose demostrado su eficacia en el aprendizaje. Sin embargo, el principal problema de estas herramientas es la falta de mecanismos para llevar a cabo un análisis cuantitativo de las habilidades adquiridas tras la simulación, ya que únicamente el 5-15 % de las personas con discapacidad tienen acceso a dispositivos de asistencia personal [24].

Para poder llevar a cabo satisfactoriamente un programa de entrenamientos de ayuda a personas discapacitadas, es necesario el desarrollo

de una serie de ejercicios estandarizados. La Organización Mundial de la Salud proporciona unos paquetes de entrenamientos de carácter básico [28] e intermedio [29], que han sido utilizados como fuente de inspiración a la hora de seleccionar los ejercicios empleados en este simulador. De los proporcionados se han elegido transferencias (ejercicio 1), movilidad (ejercicio 2) y juegos (ejercicio 3).



Figura 1: Ejercicios estandarizados para personas discapacitadas

El desarrollo de un simulador de estas características implica la utilización de usuarios reales con los que realizar las pruebas necesarias para determinar el grado de satisfacción y eficacia de los ejercicios propuestos. Sin embargo, trabajar con este tipo de usuarios requiere de una normativa específica respecto al uso de datos de carácter personal [30]. Además si se necesita la colaboración de pacientes como potenciales usuarios, hay que tener en cuenta la normativa española sobre los derechos de las personas con discapacidad [31]. Debido a la complejidad de este marco regulador, los ensayos quedan fuera del alcance de este proyecto, aunque si se ha trabajado con voluntarios, garantizando en todo momento las medidas básicas de seguridad cumpliendo la normativa WC19 para sillas de ruedas como vehículos de motor (aún si cabe son más restrictivas ya que la plataforma de movimiento utilizada cuenta con una silla de coche real que tiene un cinturón de seguridad de tres puntos) [32].

1.3. Motivación del proyecto

El proyecto surge como continuación al proyecto PhyMEL-WS (Wheelchair Simulator) [2], [3]. Este proyecto consistía en un simulador de silla de ruedas con un objetivo múltiple en función de la población a la que estaba orientado:

- 1) Por un lado para el público en general, para concienciar de los problemas a los que se enfrenta en su día a día una persona en silla de ruedas en entornos urbanos, como por ejemplo rampas, pasillos estrechos, etc.
- 2) Para las personas con discapacidad como entrenamiento para ejercitarse en el manejo de una silla de ruedas.
- 3) Para el personal médico para evaluar las habilidades de conducción de una persona en silla de ruedas y poder prescribir uno u otro modelo ajustado a sus características.
- 4) Para diseñadores y arquitectos para analizar la accesibilidad de sus modelos de edificio antes de su construcción.

Este trabajo parte del desarrollo anterior y va un poco más allá, tratando de monitorizar no solo el movimiento de la silla en el escenario sino también el movimiento del usuario en la silla. Esta herramienta otorga un sinfín de posibles aplicaciones tanto médicas como lúdicas.

Para algunas funcionalidades, especialmente para el entrenamiento de personas con discapacidad, es importante monitorizar el movimiento del usuario. El integrar un detector de movimiento, dota al simulador de un gran abanico de posibles aplicaciones a las que poder orientar su desarrollo, todas con el objetivo principal de lograr un impacto sobre la satisfacción durante la experiencia y los resultados de aprendizaje en personas discapacitadas y/o con dificultades motoras.

Ser capaz de reflejar en un entorno virtual el movimiento de una persona en la vida real supone un gran avance para el simulador porque permite aumentar las posibilidades de interacción del usuario con el simulador, consiguiendo que experimente sensaciones más próximas a la realidad [1][4]. Esto se consigue haciendo uso de ejercicios como:

- Transiciones silla de ruedas – cama.
- Recogida de objetos.
- Movimientos guiados siguiendo una trayectoria circular.

La implementación de estos ejercicios permitirá a los usuarios que los ejecuten mejorar problemas de motricidad y flexibilidad en las extremidades superiores al mismo tiempo que se familiarizan con la realización de actividades cotidianas para un discapacitado.

1.4. Objetivos

El objetivo principal de este proyecto es el desarrollo de un simulador que, mediante una plataforma de movimiento 3DOF y un sistema de monitorización (ASUS XtionPro LIVE [5]), permita al usuario de silla de ruedas entrenarse en la realización de tareas cotidianas como acostarse, recoger una habitación o hacer ejercicios específicos para fortalecer su musculatura.

Este objetivo general se ha desglosado en objetivos funcionales y técnicos a los que se ha asignado un identificador específico para relacionarlos posteriormente con los requisitos del sistema.

Entre los objetivos funcionales se pueden destacar los siguientes:

- **OF-01: Monitorizar el movimiento del usuario en la silla de ruedas.** Este objetivo consiste en ampliar la funcionalidad del simulador PhyMEL-WS desarrollado en Unity-3D para monitorizar no sólo el movimiento de la silla de ruedas en un escenario 3D sino también el movimiento real del usuario sobre la misma.

- **OF-02: Recrear entornos de interior para probar la funcionalidad del sistema de monitorización.** Será necesario diseñar varias habitaciones del interior de una casa (salón, cocina, dormitorio) donde se puedan recrear las escenas cotidianas en las que se quiere entrenar al usuario.
- **OF-03: Mejorar la sensación de inmersión del usuario mediante una visualización 3D de la escena.** Será necesario probar diferentes sistemas de visualización para ver cuál de ellos ofrece una mayor sensación de inmersión.
- **OF-04: Desarrollar una batería de ejercicios de ejemplo.** Será necesario crear varios ejercicios tipo que permitan ilustrar tanto la capacidad técnica del simulador como sus potenciales aplicaciones.

Para conseguir estos objetivos funcionales es necesario cumplir los siguientes objetivos técnicos:

- **OT-01: Comunicación entre Asus Xtion⁴ y Unity3D:** Para lograr este objetivo es necesario estudiar como manipular desde Unity3D las APIs proporcionadas por el desarrollador de Asus Xtion. Para ello será necesario estudiar paquetes de software intermedios como el kit de desarrollo Zigfu⁵, que permitan la comunicación entre ambos.
- **OT-02: Sincronización del movimiento del avatar en el mundo virtual con el del usuario.** Este es uno de los objetivos clave en el desarrollo del proyecto y requiere a su vez cumplir 3 subobjetivos que implican la adaptación de la escena, adaptación de la lógica que rige el movimiento del avatar y ajuste de parámetros generales del prefab⁶ que incluye tanto el modelo del personaje como los scripts asociados.

⁴ http://www.asus.com/es/Multimedia/Xtion_PRO_LIVE/

⁵ <http://zigfu.com/>

⁶ Recurso que permite definir las propiedades de un objeto e instanciarlo múltiples veces

- **Localización del usuario en la escena creada.** Consiste en realizar las modificaciones necesarias en la escena y configurar las opciones necesarias para que la ubicación del personaje en el escenario sea lo más precisa posible, evitando detecciones duplicadas o pérdidas de tracking.
 - **Adaptación del fichero encargado de la lógica del movimiento.** Consiste en adaptar la lógica de movimiento del personaje uniendo el movimiento de la silla con el movimiento del usuario sobre la misma, teniendo en cuenta los puntos de control y el sistema de seguimiento, y asociar el fichero de movimiento resultante al prefab del personaje.
 - **Configuración y ajuste de parámetros globales del personaje:** Consiste en utilizar pruebas de campo para configurar una serie de variables de rotación y amortiguamiento en el fichero del prefab del personaje virtual para que refleje de la forma más realista posible la interacción del usuario con el dispositivo de detección de movimiento
-
- **OT-03: Integración de nuevas escenas y modelos externos en el escenario preexistente en Unity-3D.** Consiste en añadir al escenario inicial con el que se contaba, una serie de modelos en tres dimensiones que sirvan al usuario para poder realizar los ejercicios de testeo de la monitorización de su movimiento. Todo el movimiento del personaje principal será controlado por el usuario a través del movimiento real de su cuerpo. La inclusión de nuevas escenas implicará desde el punto de vista técnico realizar transformaciones sobre el modelo como escalado o rotación, añadir o eliminar elementos de la escena y sobre todo añadir rigidez a ciertas partes del modelo para dar realismo a las colisiones de la silla sobre el entorno.
-
- **OT-04 Integración en el demostrador de gafas Oculus Rift**
Consiste en enviar al usuario la perspectiva del entorno en el que está realizando el ejercicio al display de unas gafas de realidad virtual

(Vuzix⁷ y Oculus Rift⁸). Este objetivo incluye que el simulador funcione con el display anterior (Vuzix) y la integración de un nuevo modelo de gafas (Oculus Rift). Esto mejora mucho la experiencia del usuario, ya que no tiene la necesidad de estar continuamente fijando la vista a la pantalla del ordenador para ver el modelo que está monitorizando, únicamente tendría que estar en el rango de actuación del detector de movimiento.



Figura 2: Oculus Rift y Vuzix

Oculus src: <http://blogs-images.forbes.com/erikkain/files/2014/03/OculusRift1.jpg>

Vuzix src: http://www.slashgear.com/wp-content/uploads/2008/11/vuzix_av230xl_1.jpg

- **OT-05: Sistema de guiado y control de resultados de ejercicios.**
Este objetivo consiste en añadir a los escenarios diferentes puntos de control y mecanismos de feedback que permitan al usuario identificar el siguiente objetivo a conseguir e identificar visualmente los retos ya superados.

Además de estos objetivos técnicos y funcionales, hay una serie de objetivos globales que resultan necesarios cumplir para el desarrollo satisfactorio del proyecto:

⁷ <http://www.vuzix.com/>

⁸ <http://www.oculusvr.com/>

- Estudiar la situación actual de las diferentes herramientas de aprendizaje en competencias en el mercado, para así poder tener una base de apoyo a la hora de tomar decisiones en el desarrollo de la lógica base.
- Familiarización y manejo de los dispositivos hardware a utilizar (plataforma de movimiento SMK3DOF, sensor de movimiento Xtion, gafas de realidad virtual Vuzix y Oculus Rift)
- Aprendizaje de las herramientas software (Unity3D, SketchUp y Blender) y kits de desarrollo como Zigfu y APIs asociadas a los diferentes elementos hardware.
- Aprendizaje de los lenguajes de programación C# y JavaScript.
- Evaluación de las capacidades del simulador desarrollado y comprobación del cumplimiento de todos los objetivos planteados en este apartado.

Todo el desarrollo se guiará por unos principios de diseño que se incluirán posteriormente como requisitos no funcionales y que son los siguientes:

- **Hacer una arquitectura modular y extensible:** La estructura interna de los ficheros implicados en la detección de movimiento deberá ser modular, permitiendo así integrarlos en otras escenas o proyectos y añadir posibles funcionalidades extras con facilidad.
- **Diseño sencillo e intuitivo:** El diseño del simulador debe ser intuitivo tanto para el usuario final como para el desarrollador. El objetivo es que el usuario final sea capaz de utilizarlo sin requerir de un proceso de formación previa y que el desarrollador pueda

modificar fácilmente los contenidos en función de sus necesidades o preferencias.

1.5. Estructura de la memoria

En el siguiente apartado se muestran los capítulos que componen la memoria y un resumen de los aspectos más relevantes del simulador que se analiza en cada uno de ellos.

Capítulo 1: Introducción y Objetivos

Muestra una visión global del tema sobre el cual se va a sustentar el proyecto, cuál ha sido la motivación que ha provocado su elección y qué objetivos se han marcado para desarrollar.

Capítulo 2: Estado del Arte

Presentación de algunos de los diferentes simuladores orientados a la monitorización del movimiento más relevantes a nivel mundial. Se analizan su estado actual, tanto sus características como sus aplicaciones, y se establecen unas conclusiones. Además se describen las tecnologías empleadas para el desarrollo y cuál ha sido su papel en este proyecto.

Capítulo 3: Diseño del Sistema

En este apartado se presenta y describe el trabajo de diseño que se ha llevado a cabo en este proyecto y los elementos que componen el sistema. Se hace un análisis tanto del desarrollo software como de la integración hardware

que ha sido necesaria. Además se enumeran una serie de requisitos para la construcción del simulador.

Capítulo 4: Batería de Pruebas

Recoge el proceso de aceptación de los requisitos del simulador definidos en el apartado de diseño del sistema especificando cuales han sido los recursos empleados para ello. Por último se recogen una serie de requisitos que debe cumplir el ordenador en el que se ejecuta la aplicación.

Capítulo 5: Planificación del proyecto

Muestra la metodología empleada para llevar a cabo la realización del proyecto. Incluye las etapas y tareas en las que se ha subdividido el desarrollo del mismo, así como el tiempo dedicado a cada una de ellas. Además se explica el plan de trabajo que ha servido de referencia para llevar un control del grado de evolución en el que se encontraba el simulador en todo momento. Finalmente se muestra el coste que ha supuesto este proyecto a partir de un desglose del presupuesto estimado para su realización.

Capítulo 6: Resultados y Conclusiones

Redacción de los objetivos finales alcanzados y comparación con los propuestos inicialmente. También se analizan cuáles han sido las principales conclusiones alcanzadas al desarrollar este proyecto.

Capítulo 7: Futuras Líneas de Trabajo

En este apartado se enumeran una serie de posibles ampliaciones para continuar con el desarrollo del sistema.

Capítulo 2

Estado del Arte

En este capítulo se va a realizar un análisis de los desarrollos realizados desde el punto de vista de la tecnología de vanguardia.

2.1. Introducción

En este apartado se describen los conceptos básicos necesarios para comprender el resto de la memoria y tener una visión global de los temas abordados en el proyecto.

Simulación

La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias para el funcionamiento del mismo. [6]

Monitorización

Consiste en el seguimiento en tres dimensiones a través de la captura de imágenes con información de profundidad. Es una práctica cada vez más usada en el sector de la medicina, especialmente para el tratamiento de problemas motrices y en rehabilitación de pacientes.

Realidad virtual

La realidad virtual es una ciencia basada en dispositivos tecnológicos cuyo fin es producir una apariencia de realidad que permita al usuario tener la sensación de estar presente en ella. La capacidad de interacción con un entorno simulado con total seguridad resulta un aspecto determinante para que esta tecnología se esté imponiendo en ramas como el ocio y la educación, ya que resultan más atractivas para quien las usa otorgándoles una experiencia más satisfactoria (mayor realismo implica mayor satisfacción).[7], [1]

En relación con el ambiente o entorno simulado desarrollado, la realidad virtual puede clasificarse en inmersiva y no inmersiva [8]:

- **Realidad virtual inmersiva:** está ligada a un ambiente tridimensional creado por ordenador al que se añaden dispositivos externos como cascos o guantes para capturar posición y rotación del usuario, generando una mayor sensación de integración en el entorno.
- **Realidad virtual no inmersiva:** hace uso de un ordenador y de medios que nos ofrece Internet para interactuar con personas en espacios y ambientes que no existen realmente sin la necesidad de dispositivos hápticos⁹ adicionales al ordenador.

⁹ Relativo al sentido del tacto

2.2. Análisis de los simuladores de apoyo a discapacitados existentes

A lo largo de los últimos años han ido apareciendo prototipos de simuladores para rehabilitación de pacientes en silla de ruedas. A continuación, se analizará algunos de ellos, indicando en qué fase de desarrollo se encuentran y analizando comparativamente sus principales características.

Simulador TOYRA

El proyecto Toyra (Terapia objetiva y rehabilitación audiovisual) surgió en 2011 para mejorar las terapias de rehabilitación. Las instituciones participantes en su desarrollo fueron:

- Indra
- Fundación Rafael del Pino
- Fundación para la Investigación y la Integración del Hospital Nacional de Paraplégicos de Toledo.

Este producto está orientado a la rehabilitación de tren superior (tronco y extremidades superiores) a través de un entorno virtual capaz de detectar el movimiento del paciente. Para ello utiliza sensores de captación de movimiento y posteriormente envía al sistema su localización, pudiendo simular en el entorno virtual el movimiento real que realiza el paciente. Toyra evalúa, registra y analiza la información recogida y genera unos resultados que permite a los rehabilitadores llevar un control del grado de progreso del usuario.

Toyra se compone a su vez de dos subsistemas que permiten la gestión y centralización de todo el proceso de recogida de información [9]:

- **Subsistema de Gestión de Terapia Objetiva (servidor central).**
Recoge, centraliza y gestiona toda la información relevante al paciente.
- **Subsistema de Terapia Objetiva Interactiva (Estaciones de Terapia).** Permite desplegar estaciones de terapia en función de las necesidades del centro. Actualmente existen dos versiones: una para ámbito hospitalario y otra para aplicación ambulatoria o domiciliaria.

Además, Toyra proporciona dos modalidades distintas (personal médico y paciente), cada una de ellas con unas especificaciones individuales en función del ámbito al que está orientada su utilización:

- **Toyra Asistido:** para el ámbito profesional incluyendo un registro objetivo de medidas, análisis de resultados y seguimiento del tratamiento adoptado. [10]
 - **Especificaciones:**
 - ❖ Servidor
 - ❖ Pantalla de paciente
 - ❖ Estación de trabajo
 - ❖ Kit de sensores
 - ❖ Software hospitalario
- **Toyra Autónomo:** para el paciente, permitiéndole llevar a cabo el tratamiento en casa con un seguimiento a distancia. [11]
 - **Especificaciones:**
 - ❖ Conexión a Internet
 - ❖ Kinect
 - ❖ Monitor de Televisión con HDMI
 - ❖ Software online
 - ❖ Documentación usuario y set de ejercicios



Figura 3: Simulador Toyra

src: http://www.toyra.org/00REPO/PROV/chica_pq1.jpg

Este simulador se encuentra en una fase de desarrollo todavía precoz, sin embargo su crecimiento es notorio ya que en febrero de este mismo año se han instalado simuladores Toyra en los hospitales ASPAYM (Valladolid), Centro de Referencia Estatal San Andres de Rabanedo (León), Hospital de la Beate (Madrid) y Complejo hospitalario de Toledo y Hospital Nacional de Paraplégicos (Toledo). Además se han ampliado las patologías a las que ayudarán a tratar (Parálisis cerebral, Ictus, Traumatismo craneoencefálico, etc.)

Simulador de Clemson y DriveSafety (DriveSafety CDS-250W)

Este simulador surgió en 2013 como resultado de la colaboración de la universidad de Clemson y la empresa DriveSafety, que proporcionó la tecnología para la simulación. El objetivo de este proyecto es el desarrollo de un simulador de conducción con adaptabilidad a silla de ruedas diseñado para rehabilitar a pacientes, que está siendo utilizado por el ejército y la marina.

Este simulador cuenta con un asiento base extraíble y elevadores motorizados para adaptarse a cualquier tamaño de silla de ruedas. También viene con un conjunto de controles manuales, mandos giratorios, y pedales.

Todo esto unido a una extensa biblioteca de escenarios de conducción para el tratamiento clínico otorga a este producto un gran valor. Sin embargo no cuenta con un sistema de monitorización del usuario, lo que impide llevar un control automatizado del movimiento del paciente y de su grado de mejora. [14]

El objetivo final de este simulador es la rehabilitación de habilidades cognitivas, perceptivas y físicas a través de la práctica de habilidades de conducción realistas en un ambiente seguro. Además, al incluir en el proyecto a investigadores de la rama de la psicología, también buscan estudiar factores como la habilidad de conducción y las limitaciones en personas de la tercera edad.



Figura 4: Simulador DriveSafety CDS-250W desarrollado por la Universidad Clemson

src: <http://www.redorbit.com/media/uploads/2011/06/03cc39b1b1b7500a803c970c320bdcbbc-617x410.jpg>

El equipo de desarrollo pretende que en el futuro, los datos recogidos por esta herramienta permitan desarrollar vehículos más seguros y eficientes, así como dispositivos que permitan conducir a personas con discapacidades.

Simulador del CIDIF

Otro ejemplo de simulador para la ayuda de personas con problemas de movilidad es el que está siendo desarrollado actualmente (2014) por el CIDIF (Centro de investigación en discapacidad física) de la fundación Aspaym de Castilla y León.

Esta herramienta capta el movimiento del paciente, permitiendo optimizar los ejercicios de rehabilitación de las extremidades superiores. Para ello se apoyan en una serie de sensores inerciales con los que recoger la información del movimiento del paciente en tiempo real para su posterior análisis.



Figura 5: Simulador para pacientes con ictus desarrollado por CIDIF

src: <http://www.europapress.es/castilla-y-leon/noticia-cidif-aspaym-investiga-mejoras-movilidad-pacientes-ictus-realidad-virtual-20140205143550.html#!kalooga-13936/~valladolid%20~juan%5E0.75>

El objetivo de este simulador virtual es el de mejorar la movilidad en los brazos en aquellas personas que hayan sufrido un ictus, patología que se ha convertido en los últimos años en la primera causa de muerte entre las mujeres

y la segunda entre los hombres, además de causar infinidad de dolencias, problemas motores y cognitivos en adultos.

El proyecto se encuentra todavía en una fase embrionaria y está pendiente de evaluación. Uno de los elementos a evaluar será el grado de satisfacción del paciente. Inicialmente trabajarán con una muestra de entre 15-20 pacientes para determinar los ejercicios más productivos y a partir de ellos determinar los resultados. [15]

Además de este simulador, CIDIF cuenta con muchos otros proyectos orientados al diagnóstico y tratamiento de las lesiones del hombro y terapias de vibraciones y a la electroestimulación sobre el sistema vascular y músculo-esquelético en pacientes con lesiones medulares.

PUCP – Motion simulator

Como alternativa a los métodos de rehabilitación tradicionales, un grupo de docentes de ingeniería Mecánica y Electrónica de la Pontificia Universidad Católica de Perú (PUCP) junto a alumnos trabajaron en el desarrollo de una herramienta capaz de mejorar las habilidades motrices de pacientes con problemas en los miembros inferiores. En 2005 comenzaron a dar los primeros pasos orientados a la creación de un simulador que permitiera al usuario realizar movimientos coordinados y suaves, simulando la marcha en un entorno virtual. En 2009 se inició la construcción gracias a la financiación del proyecto por parte del FINCyT (Fondos para la Innovación, Ciencia y Tecnología) y el apoyo del Hogar Clínica San Juan de Dios.

El simulador cuenta con el VICON Motion Capturing System, que obtiene imágenes tridimensionales para analizar el movimiento con el fin de generar diagnósticos clínicos y tratamientos personalizados. Además permite, a través de Internet, la gestión remota de la información generada. De forma que se pueda brindar un servicio de telerehabilitación en centros de salud alejados. [16]

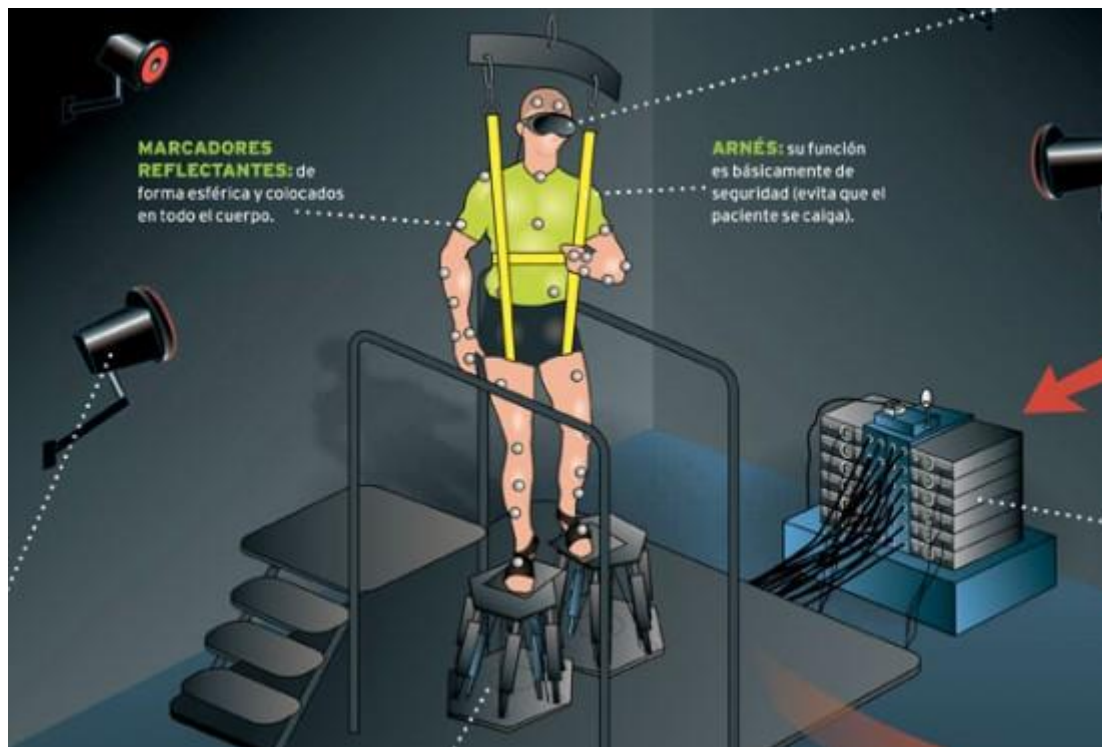


Figura 6: Simulador desarrollado en la PUCP

src: <http://www.dicyt.com/viewItem.php?itemId=29068>

La intención de los desarrolladores es posicionarlo en el sistema de salud de Perú y convertirlo así en una alternativa viable a los sistemas tradicionales de rehabilitación para personas discapacitadas.

2.3. Conclusiones

Tras analizar los simuladores mencionados se puede concluir que la mayoría de los que se están desarrollando en la actualidad tienen como objetivo primordial el proporcionar al usuario una sensación de inmersión en un mundo virtual que permita monitorizar las interacciones entre el paciente y el entorno de simulación.

El principal déficit que presenta este tipo de simuladores es que el usuario no dispone de una plataforma de movimiento que simule el movimiento

que tendría el paciente en silla de ruedas en un entorno real. Además casi todos usan monitores como displays, excepto PHYMEL y PUCP que usan además HMD.

Tareas cotidianas como llenar un vaso de agua o recoger un objeto del suelo pueden resultar complejas para una persona con discapacidad porque la información que está recibiendo del entorno unido a su discapacidad hacen que no disponga en muchos casos de la habilidad suficiente para llevar a cabo la acción sin producirse situaciones desagradables como caídas de la silla o dolores en la espalda por estiramientos excesivos de los brazos.

El simulador que se está desarrollando está basado en una plataforma de movimiento 3DOF y un dispositivo de detección de movimiento, al que se le ha incorporado una gafas de visión en tres dimensiones para dar más realismo si cabe. El sistema permitirá por un lado realizar ejercicios en el que el usuario controlará la silla de ruedas por medio de un Joystick o teclado, y por otro realizar ejercicios en el que el usuario controlará al personaje de la silla a través del movimiento de su tren superior.

A continuación, en la siguiente tabla se realiza una comparación entre las características de los simuladores actualmente en el mercado, que han supuesto la principal fuente de inspiración para el desarrollo de este proyecto.

Tabla 1: Resumen de los proyectos científicos / simuladores estudiados anteriormente

Proyecto	Año	País	Características	Info. Feedback	Plataforma de movimiento	Interfaz visual	Modo de funcionamiento	Control de usuario	Tracking de usuario
Toyra (Asistido)	2011	Indra, Fundación Rafael del Pino, Fundación para la Investigación y la Integración del HNPS, España.	<ul style="list-style-type: none"> - Registro de medidas - Análisis de resultados - Seguimiento del tratamiento - Estación de trabajo 	Visual / Oral (rehabilitador)	No	Monitor de televisión estándar	Activo	Joystick / Monitorizado	Sí (Sensores inerciales)
Toyra (Autónomo)	2011	Indra, Fundación Rafael del Pino, Fundación para la Investigación y la Integración del HNPS, España.	<ul style="list-style-type: none"> - Fácil manejo - Tratamiento remoto - Soporte telefónico - Formación online 	Visual	No	Monitor de televisión estándar (con HDMI)	Activo	Monitorizado	Sí (Kinect)

PUCP – Motion Simulator	2009	Docentes y alumnos de la Pontificia Universidad Católica de Perú, Perú	<ul style="list-style-type: none"> - Registro de imágenes 3D - Análisis de resultados - Telerehabilitación personalizada 	Visual	No (Estación de ayuda)	HMD ¹⁰	Activo	Monitorizado	Sí (ICON Motion Capturing System)
DriveSafety CDS-250W	2013	Universidad de Clemson, DriveSafety, EE.UU.	<ul style="list-style-type: none"> - Asiento extraíble - Elevadores motorizados - Sistema de reproducción de audio - Campo visual de 110 grados 	Visual	No (plataforma de conducción)	Monitor de televisión	Activo	Volante y pedales	No
Physiobot	2014	CIDIF (Fundación ASPAYM Castilla y León), España.	<ul style="list-style-type: none"> - Registros de información - Análisis de resultados - Búsqueda de los ejercicios más productivos 	Visual	No	Monitor de televisión estándar	Activo	Monitorizado	Sí (sensores inerciales)
PhyMEL-WS Wheelchair Simulator	2014	Universidad Carlos III, Madrid España	<ul style="list-style-type: none"> - Visión en 3D (HMD) - Plataforma 3DOF - Fácil exportación e instalación 	Visual	Sí (3DOF)	HMD / monitor estándar	Activo / Pasivo	Teclado / Joystick / Monitorizado	Sí (Asus Xtion)

¹⁰ Head-mounted display

2.4. Tecnologías empleadas

En el siguiente apartado se hará un análisis de las herramientas hardware y software que se han utilizado para la realización del proyecto

1. **Unity3D PRO v.4.3.4f1**
2. **SketchUp**
3. **Blender**
4. **ASUS XtionPro Live**

2.4.1. Unity3D PRO v.4.3.4f1



Figura 7: Logo de Unity

src: <http://img2.meristation.com/files/imagenes/general/unity3d1.jpg>

Unity3D (Unity (software)¹¹) es un motor de videojuegos disponible para un gran número de plataformas tales como Windows, OSX, Linux, Xbox 360,

¹¹ <http://unity3d.com/es>

PlayStation 3, Playstation Vita, Wii, Wii U, iPad, iPhone, Android y Windows Phone. Además gracias a su Plug-in Web, también permite realizar juegos para navegadores, para Windows y para Mac.[17]

Fue creado por Unity Technologies con el principal objetivo de crear un motor gráfico asequible a todos, “democratizar el desarrollo de juegos”. La primera versión de Unity se lanzó en 2005 en la Conferencia Mundial de Desarrolladores de Apple y obtuvo el éxito necesario para continuar con su evolución hasta convertirse en uno de los principales motores de desarrollo de videojuegos utilizados en la actualidad.

Una de las principales ventajas con las que cuenta Unity, es el amplio rango de compatibilidad que posee, ya que soporta integración con 3ds Max, Maya, Softimage, Blender, Modo, Zbrush, Cinema 4D, Cheetah 3D, Adobe Photoshop, Adobe Fireworks y Allegorithmic. Además posee un sistema de scripting vía Mono que permite la depuración del script mediante MonoDevelop (herramienta que permite programar en C#, JavaScript y Boo).

Durante el proyecto se comenzó trabajando con la versión 4, pero al lanzar la versión 4.3 durante el desarrollo, se tuvo que actualizar el software. Esta modificación no ha supuesto un problema relevante ya que los cambios entre versiones han sido mínimos (cabe destacar el ajuste “Dopesheet” para animaciones basadas en fotogramas clave en la ventana de animación para animar sprites¹² como alternativa a la tecnología “Mecanim” introducida en la versión 4 que se basaba en la creación de máquinas de estados y árboles de mezcla desde editor). Se ha requerido trabajar con la versión Pro de esta última versión debido a un problema de compatibilidad de la versión gratuita con la integración de las Oculus Rift.

¹² Mapa de bits

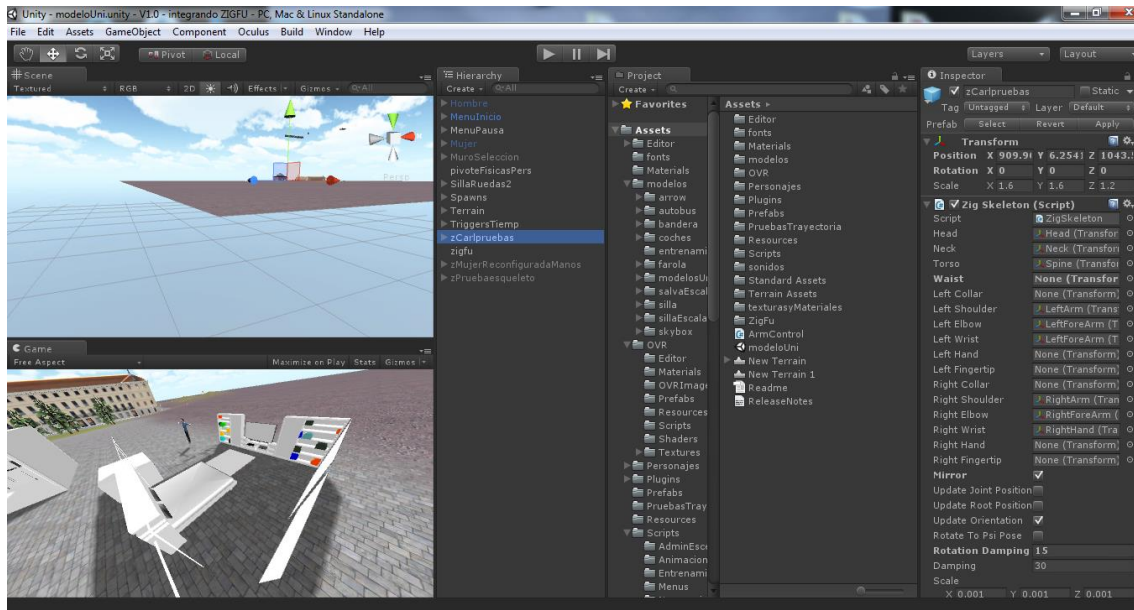


Figura 8: Entorno de trabajo de Unity3D

Las principales funcionalidades con las que cuenta esta herramienta de desarrollo son:

- Crear juegos para diferentes plataformas
- Realidad virtual
- Recorridos virtuales

Se puede considerar a Unity3D la piedra angular sobre la que ha girado el desarrollo de todo el proyecto. La integración del hardware, la elaboración de las escenas, las físicas de sus elementos y la programación del software se han llevado a cabo utilizando este motor gráfico y las herramientas que proporciona. Nos ha permitido recrear entornos de simulación con un alto grado de realismo que permiten al usuario visualizar el entorno tanto en primera persona como en tercera persona. La plataforma de desarrollo de Unity3D ha sido la encargada de soportar la ejecución del simulador.

2.4.2. SketchUp



Figura 9: Logo SketchUp

src : http://www.apaspa.com/img_db/promo/Logo-sketchup.jpg

SketchUp es un programa para modelado en 3D y diseño gráfico en entornos de arquitectura, ingeniería civil, diseño industrial y videojuegos entre otros. Desarrollado por @Last Software en 2000, fue independiente hasta que en el año 2006 fue adquirida por Google como consecuencia de la gran popularidad adquirida. En el año 2012 fue vendida a Trimble, de ahí que también sea conocido como Trimble SketchUp.[18]

La principal característica de esta herramienta reside en que permite realizar diseños complejos y precisos en 3D de forma sencilla. Para ello, además de contar con una interfaz de manejo intuitiva, dispone a su vez de una galería de objetos, imágenes y texturas y de videos tutoriales paso a paso que permiten al usuario utilizar el programa de manera intuitiva y flexible.

La última versión, desarrollada hasta la fecha, la 14.0.4900 (2014) ha sido la que se ha utilizado en el desarrollo del simulador. Funciona bajo Windows Vista, Windows 7 y Windows 8 y en OS X y Mac OS 10.7 o superior. Actualmente no existe versión disponible para Linux pero están trabajando para convertir a este programa en una aplicación multiplataforma al igual que lo es Unity.

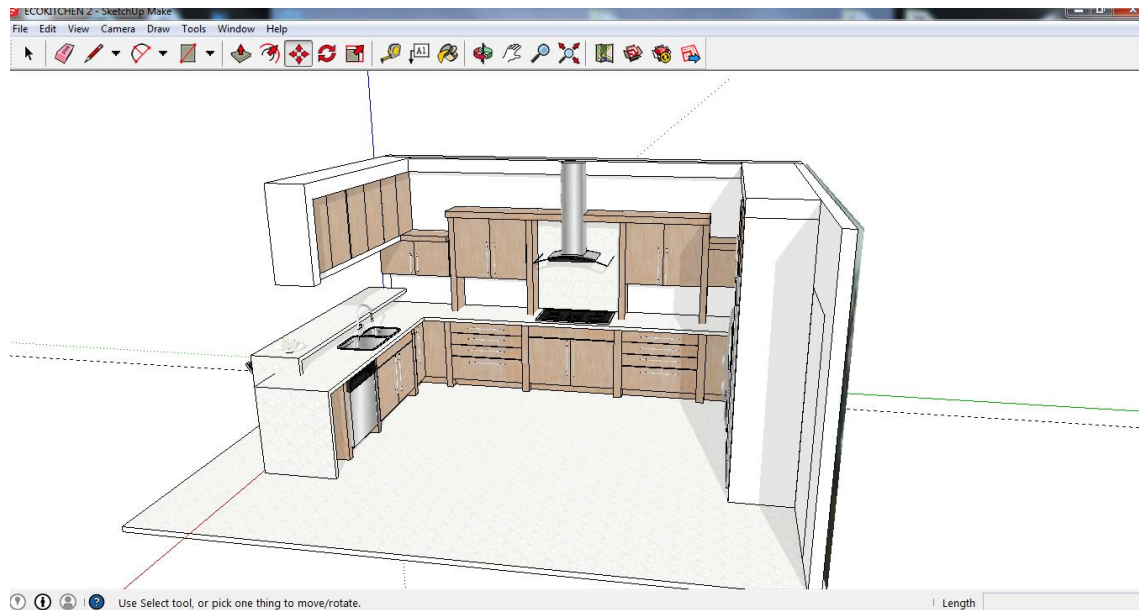


Figura 10: Entorno de trabajo de SketchUp

Las principales funcionalidades con las que cuenta esta herramienta de modelado son:

- Conceptualizar y modelar imágenes en 3D.
- Añadir localización gracias a la integración con Google Maps.
- Cálculos de volumen y área.
- Clasificador BIM ¹³(asignación de etiquetas).
- API Ruby para crear herramientas de dibujo personalizadas.

La utilidad que ha tenido este programa durante el desarrollo ha sido la de modificar los modelos obtenidos de google 3D gallery¹⁴ para utilizarlos en las escenas y la de crear el container de los modelos correspondientes a la modalidad de entrenamientos. La simplicidad de manejo ha permitido realizar pequeños cambios a los modelos utilizados de forma rápida, ya que posee una

¹³ Etiqueta la geometría con tipos de objetos estándar del sector de la arquitectura: muros, losas, forjados, etc.

¹⁴ <https://3dwarehouse.sketchup.com/?redirect=1&hl=es&hl=es>

curva de aprendizaje más rápida que otros programas de modelado 3D mucho más complejos como Autodesk 3ds Max¹⁵.

2.4.3. Blender



Figura 11: Logo Blender

src: <http://download.blender.org/institute/logos/blender-plain.png>

Blender¹⁶ es un programa gratuito y de código abierto (Open Source) utilizado para el modelado, renderizado, animación y creación de gráficos para una gran variedad de plataformas. Además posee la capacidad de desarrollar videojuegos gracias a que dispone además de un motor de juegos.

En 1988 se creó el estudio de animación NeoGeo¹⁷, que rápidamente se convirtió en uno de las casas más destacadas de animación en Europa. En 1998 derivada de NeoGeo surgió la compañía Not a Number (NaN), con la que se buscaba fomentar el mercado y desarrollar Blender teniendo como base el deseo de crear y distribuir un producto gratuito y multiplataforma, concepto revolucionario para la época (los programas comerciales costaban miles de dólares). Sin embargo el escaso número de ventas y las dificultades financieras provocaron que los inversores que sostenían económicamente la empresa dejaran de hacerlo, llevando a la desaparición de Nan. En 2002 Blender fue liberado bajo los términos de Licencia Pública General de GNU (GLP) con la

¹⁵ <http://www.autodesk.es/products/autodesk-3ds-max/overview>

¹⁶ <http://www.blender.org/>

¹⁷ <http://www.neogeo.es/>

ayuda de la organización no lucrativa Blender Foundation creada unos meses antes.[19]

Además de ser un producto multiplataforma, libre y gratuito, Blender tiene una serie de características que lo hacen muy atractivo para animadores independiente y pequeños estudios (sobre todo anuncios de televisión):

- Capacidad para una gran variedad de primitivas geométricas (curvas, mallas poligonales, vacíos, etc.).
- Incluye herramientas de cinemática inversa, deformaciones por armadura o cuadrícula y partículas estáticas o dinámicas.
- Permite edición interna de audio y sincronización de video.
- Motor de juegos 3D integrado.
- Capacidad para realizar match moving (técnica de efectos visuales que permite insertar gráficos en un vídeo con la posición, escala, orientación y movimiento acorde a los objetos presentes en la escena).

Para este trabajo se ha utilizado la versión 2.69.0 (2013) de Blender, aunque en este caso la decisión sobre la versión a utilizar no fue determinante porque solo se utilizó para tareas sencillas como renderizado, edición de modelos o exportación a .fbx, y estas funcionalidades están soportadas por casi todas las versiones disponibles. La versión 2.60.0 añade la herramienta Biset, para cortar mallas por la mitad, pero que no se ha utilizado en este proyecto.

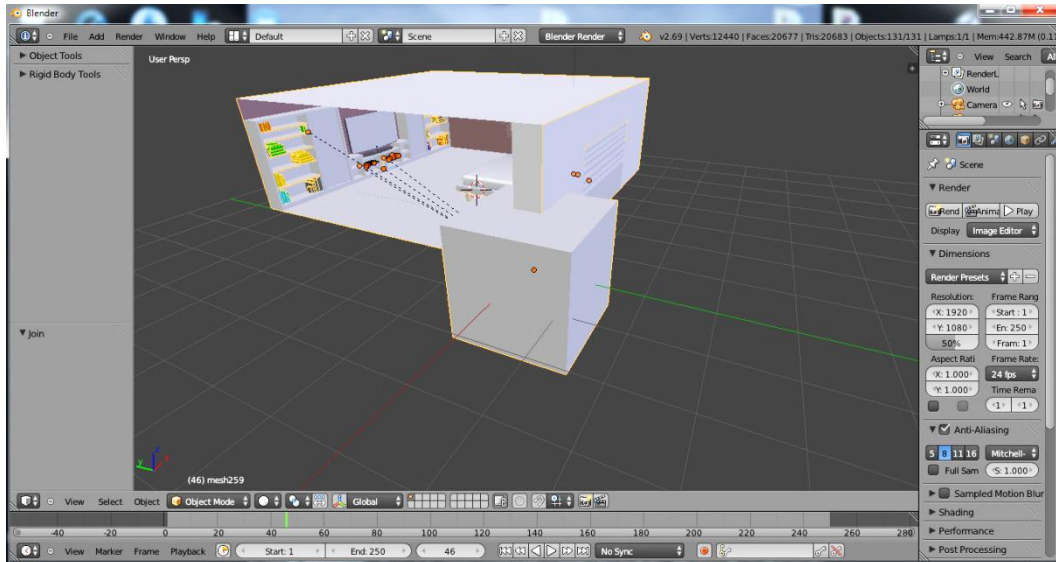


Figura 12: Entorno de trabajo de Blender

Las principales funcionalidades que posee este programa informático multiplataforma son:

- Representación fotorealista (potente motor de renderizado).
- Modelado rápido (gama completa de herramientas).
- Rigging (transformación de un modelo en un personaje).
- Herramientas de animación keyframing (fotogramas clave).
- Recuperación y edición de mallas (Unwrapping).

En este proyecto se ha hecho uso de esta herramienta principalmente para el renderizado de los modelos creados/modificados en SketchUp y para su posterior exportación al formato compatible con Unity3D (.fbx).

2.4.4. ASUS XtionPRO Live



Figura 13: Dispositivo ASUS XtionPRO LIVE

src: [http://1.bp.blogspot.com/-](http://1.bp.blogspot.com/-dJO_RPKOCtw/TiWuTgvLzRI/AAAAAAAAABA4/cZU4UsdCq5Y/s1600/xtion_pro_live_2.jpg)

[dJO_RPKOCtw/TiWuTgvLzRI/AAAAAAAAABA4/cZU4UsdCq5Y/s1600/xtion_pro_live_2.jpg](http://1.bp.blogspot.com/-dJO_RPKOCtw/TiWuTgvLzRI/AAAAAAAAABA4/cZU4UsdCq5Y/s1600/xtion_pro_live_2.jpg)

El dispositivo XtionPRO LIVE de ASUS¹⁸ es una solución para desarrollo de aplicaciones y juegos basados en la detección de movimiento. A principios de 2011 Asus lanzó su sensor Xtion Motion, diseñado para llevar el control por movimiento al estilo Kinect de Microsoft a la PC, sin embargo en comparación con las especificaciones del dispositivo Kinect, todavía tenía mucho margen de mejora. Para corregir esto, lanzó a finales del mismo año la nueva actualización de sensor de movimiento de PC, la Asus XtionPRO LIVE.[5]

Se trata de la primera solución de desarrollo software profesional en color (RGB) orientada para PC con un plug-n-play para USB, lo cual le otorga al usuario una gran comodidad a la hora de instalarla en su ordenador y comenzar a trabajar con ella. Este dispositivo se compone de una serie de sensores como cámara de infrarrojos, detectores de profundidad y tracking de usuario permitiendo a los desarrolladores crear con mayor facilidad sus aplicaciones. (*Se verá más en detalle en el apartado 3.5.1. Dispositivo de detección de movimiento*)

¹⁸ <http://www.asus.com/es/>



Figura 14: Detección de movimiento con XtionPRO LIVE

src: http://www.asus.com/Multimedia/Xtion_PRO_LIVE/overview/

El tipo de funcionalidades que permite implementar este dispositivo con los sensores mencionados son las siguientes:

- Detección de gestos.
- Seguimiento del cuerpo.
- Detección de imágenes en color.
- Reconocimiento y control de voz.

Para el desarrollo de este proyecto, este dispositivo ha sido, junto al motor gráfico, el elemento clave en torno al cual se ha desarrollado toda la lógica de entrenamientos. La detección de movimiento a partir del dispositivo Xtion ha permitido reflejar en un entorno de simulación la idea de realizar una herramienta para ayudar a aquellas personas con deficiencias físicas tanto a rehabilitarse como a corregir posturas incorrectas. El principal problema es que el dispositivo XtionPro Live de ASUS no proporciona las herramientas para trabajar en entornos de desarrollo como el que se está utilizando en este proyecto (Unity3D). Por esto se ha necesitado de un Plug-in externo a este dispositivo, ZDK de zigfu, que ha facilitado la comunicación multiplataforma Xtion-Unity3D.

A continuación, la Tabla 2 resume las principales características de cada una de estas herramientas.

Tabla 2: Resumen de las principales características de las tecnologías empleadas durante el desarrollo

Nombre	Desarrollador	Versión (Fecha)	Principales funcionalidades	Utilidad en el proyecto
Unity3D	Unity Technologies	4.3.4f1 (2013)	<ul style="list-style-type: none"> • Crear juegos multiplataforma • Realidad virtual • Recorridos virtuales 	<ul style="list-style-type: none"> • Integración hardware • Elaboración escenas • Programación software
SketchUp	Trimble (previamente @Last Software y Google)	14.0.4900 (2014)	<ul style="list-style-type: none"> • Modelado 3D • Clasificador BIM • API Ruby 	<ul style="list-style-type: none"> • Edición de modelos
Blender	Fundación Blender	2.69.0 (2013)	<ul style="list-style-type: none"> • Renderizado • Rigging • Unwrapping 	<ul style="list-style-type: none"> • Renderizado • Exportación .fbx
Asus XtionPRO	Asus	LIVE (2011)	<ul style="list-style-type: none"> • Detección gestos • Reconocimiento y control de voz • Detección audio y video 	<ul style="list-style-type: none"> • Monitorización

Capítulo 3

Diseño del Sistema

En el siguiente apartado se va a realizar un desglose del simulador desarrollado y se van a analizar sus principales componentes: hardware, software, modelado, ejercicios, etc.

3.1. Introducción

El trabajo realizado en este proyecto consiste en la integración de un dispositivo de detección de movimiento en el simulador de silla de ruedas PhyMEL-WS que permita la monitorización del movimiento del usuario en el entorno de desarrollo de Unity3D. Como ejemplo de aplicación se crearán una serie de escenarios orientados a la realización de ejercicios básicos para discapacitados. El carácter interactivo de los movimientos a realizar, que estarán guiados en todo momento, busca motivar al paciente y favorecer su correcto aprendizaje.

Una vez se entra en el sistema, el usuario dispone de tres opciones de configuración del simulador en función de la experiencia que quiera realizar. Entre las opciones principales se encuentran tres tipos de modalidades:

- Recorrido libre
- Recorrido guiado

- Entrenamientos

Todas las opciones permiten la selección libre del género del avatar (hombre o mujer) y del momento del día (día o noche). Esta última funcionalidad no resulta relevante para este estudio inicial puesto que las pruebas se realizarán en espacios interiores bien iluminados. En la figura 15 se muestra cuál es la apariencia del sistema en el momento en que se accede.[3]



Figura 15: Pantalla de acceso al sistema

Una vez que el usuario ha entrado en el sistema, éste le propone una serie de ejercicios en función de los objetivos que trate alcanzar. Para ello se han desarrollado tres ejercicios de los cuales dos hacen referencia a tareas cotidianas, recoger una cocina y tumbarse en una cama, y el otro permite mejorar la motricidad de las extremidades superiores. Cabe destacar también que los tres ejercicios están implementados en recintos interiores de forma que el usuario se encontrará obstáculos tales como sofás, mesas, sillas, etc.

En las figuras 16, 17 y 18, se ilustra la apariencia de la selección de cada uno de los ejercicios de entrenamiento:



Figura 16: Ejercicio 1 – Transición silla de ruedas/cama



Figura 17: Ejercicio 2 – Recoger objetos de una cocina



Figura 18: Ejercicio 3 – Fortalecimiento de brazos

Una vez seleccionado el ejercicio, comenzará la monitorización del movimiento del usuario y la sincronización con el avatar elegido para realizar el ejercicio. Cada ejercicio consiste en una secuencia de movimientos con un objetivo concreto. Se considera que el ejercicio ha sido completado con éxito cuando el usuario ha colocado cada parte del cuerpo (manos y pies) en las posiciones escogidas en la secuencia indicada, disponiendo para ello de todo el tiempo que desee. En todo momento un audio guía al usuario sobre la siguiente tarea a realizar y los cambios de color de amarillo a verde, indican cuando un determinado paso ha sido superado.

La plataforma de movimiento 3DOF otorgará al usuario sensaciones de percepción de la inclinación, vibración, aceleración y colisiones de la silla de ruedas.

El sistema de gestión de cámaras dispondrá en todo momento de una visión en primera persona con visión lateral (además de una secundaria que facilite la visión global de la escena), que permitirá una mayor inmersión del usuario en el caso de que se utilicen alguno de los dispositivos HMD integrados.

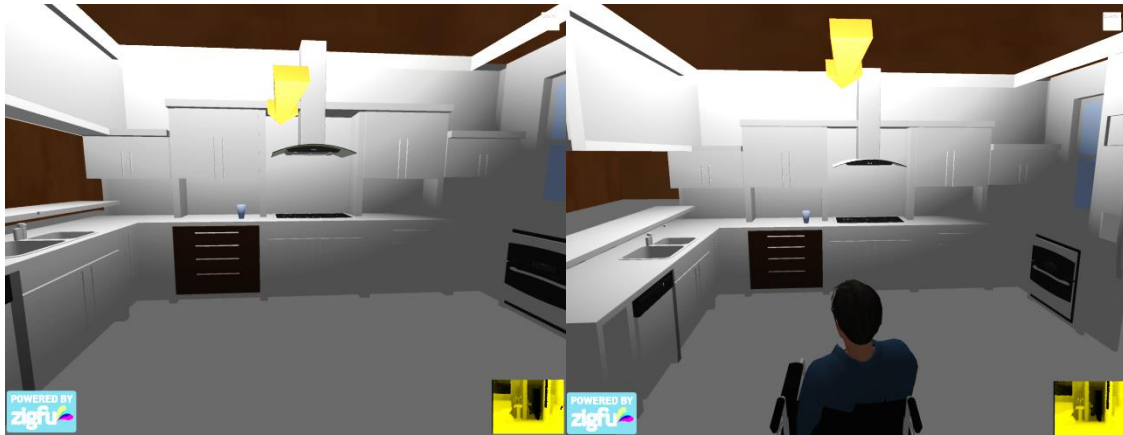


Figura 19: Visión escena en primera persona vs tercera persona

3.2. Diseño de ejercicios

Como se ha citado en el anterior apartado, el simulador cuenta con tres ejercicios disponibles para la modalidad de entrenamientos, todos con el objetivo de mejorar la motricidad de los usuarios y de familiarizarlos con situaciones reales.

Pese a tratarse de ejercicios con temáticas diferentes, todos poseen en común una estructura base que permite el desarrollo correcto del ejercicio:

- 1) Es necesaria la colocación de la silla de ruedas del usuario en una posición determinada para comenzar con la interacción con los componentes de la escena. En este caso el usuario recibe feedback tanto visual como físico. El usuario siente por ejemplo las colisiones de la plataforma con el entorno.

- 2) El usuario tendrá que coger o tocar objetos que actúan a modo de control para ir superando los pasos necesarios para completar el ejercicio. En este caso, al no disponer de un guante háptico el usuario recibirá solo feedback visual y auditivo de su interacción con el entorno.
- 3) Será imprescindible que el usuario termine de escuchar la explicación de ayuda para que se habilite el siguiente punto de control.

Primer ejercicio

El primer ejercicio se centra en mostrar al usuario los pasos necesarios a realizar para llevar a cabo la transición desde una silla de ruedas hasta una cama. Éste, se trata de uno de los movimientos más importantes que debe aprender a realizar un discapacitado y que requiere de un entrenamiento específico.

En la figura 20 se muestra un análisis detallado de los pasos a realizar para la ejecución del ejercicio.

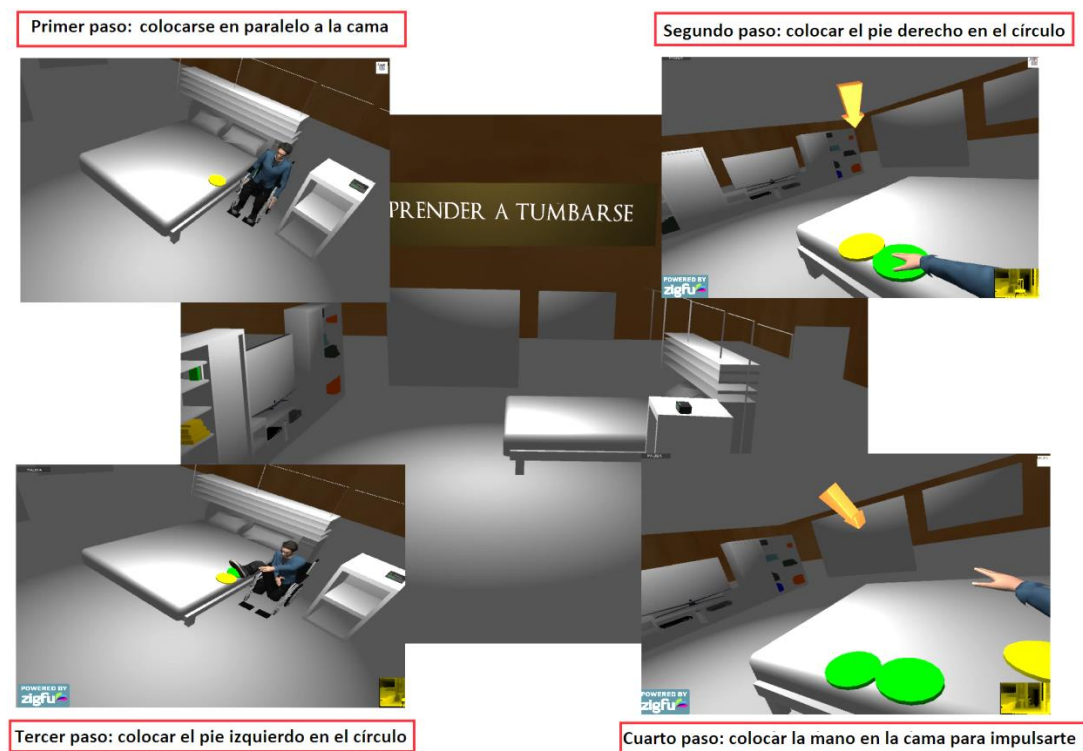


Figura 20: Desglose del primer ejercicio de entrenamientos

Segundo ejercicio

Siguiendo la línea del primer ejercicio, en el segundo se realiza una actividad cotidiana como es recoger una cocina. En él, el usuario tendrá que acercarse a determinados puntos señalados en la cocina para recoger los utensilios indicados y llevarlos al fregadero.

En la figura 21 se muestra un análisis detallado de los pasos a seguir para la realización del ejercicio.

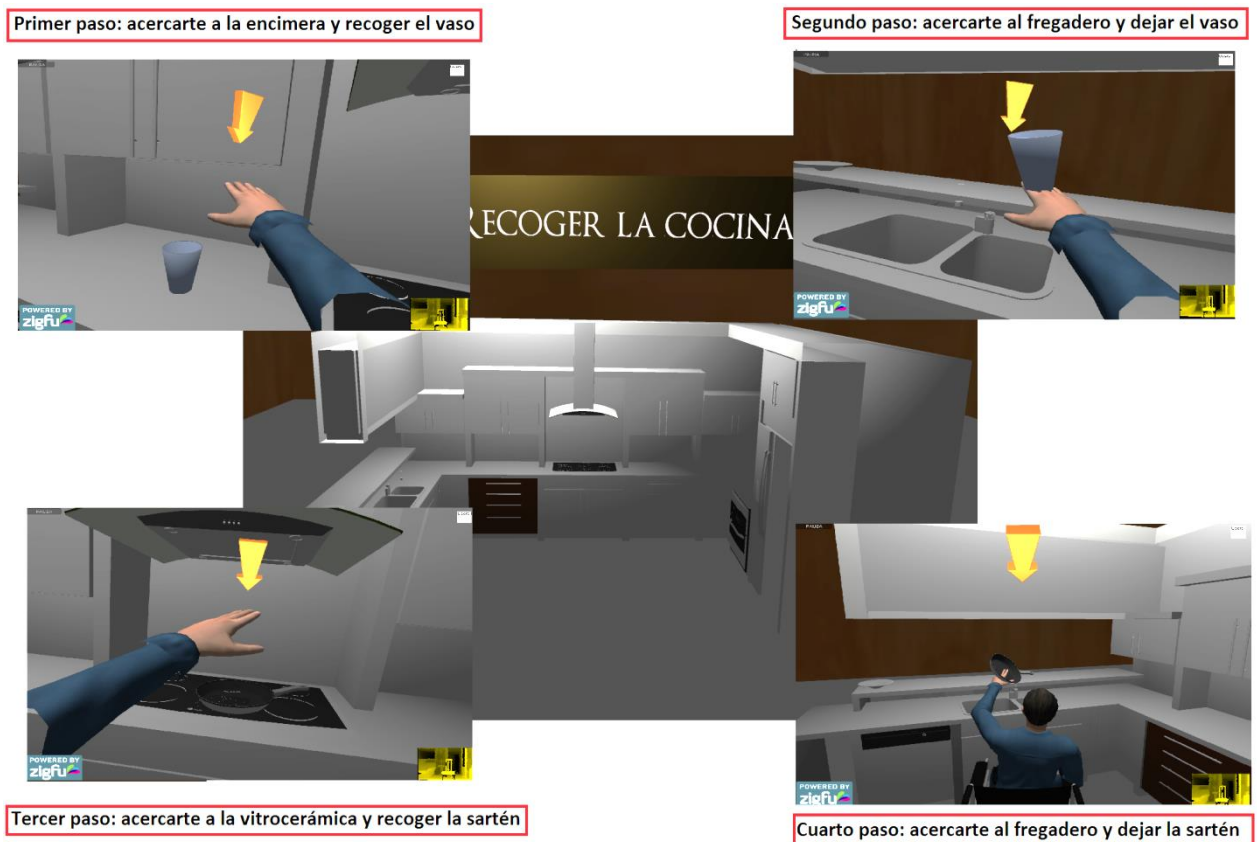


Figura 21: Desglose del segundo ejercicio de entrenamientos

Tercer ejercicio

Es el ejercicio de carácter más general. No se centra en realizar una actividad en concreto, si no en trabajar la movilidad de los brazos a través de diferentes ejercicios para mejorar la motricidad del tren superior. En él los usuarios tendrán que realizar un movimiento guiado con los brazos en forma de

trayectoria circular de forma que vaya tocando unas esferas que actúan como puntos de control.

En la figura 22 se muestra un análisis detallado de los pasos a realizar para la ejecución del ejercicio.

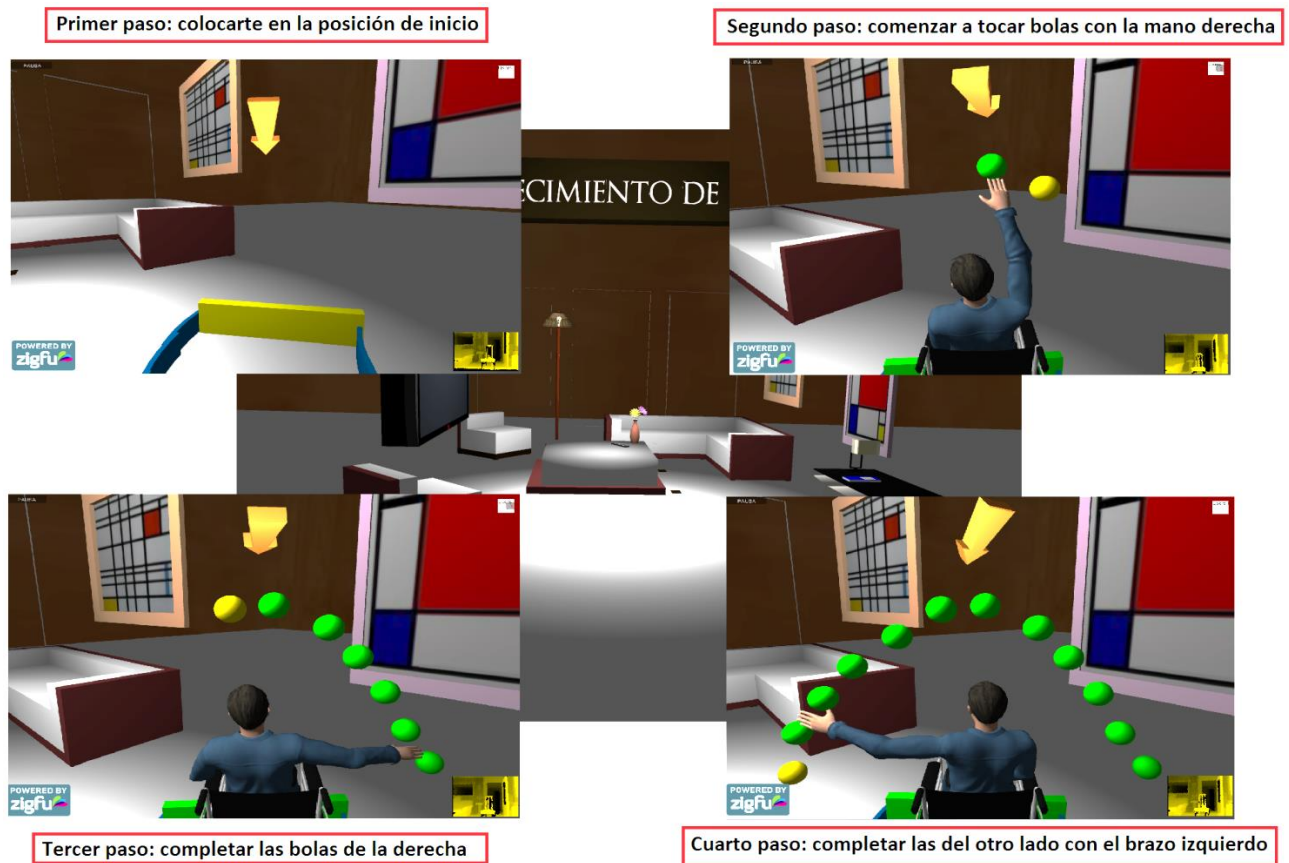


Figura 22: Desglose del tercer ejercicio de entrenamientos

3.3. Modelado y Desarrollo en Unity3D

El motor de desarrollo de Unity3D no solo actúa de núcleo de ejecución del simulador, sino que además permite la modificación interna de modelos desarrollados con otras herramientas (Blender, SketchUp, etc.) importados en el sistema. A continuación se va a realizar un análisis de los principales modelos editados y en qué ha consistido en cada caso el proceso de edición.

3.3.1. Container

Teniendo como referencia la idea de independizar los modelos necesarios para llevar a cabo la monitorización de los empleados en los recorridos libre y guiado, se empleó un elemento contenedor, que básicamente consiste en un cubo hueco. Este modelo se diseñó utilizando el programa SketchUp y fue modificado en el editor de Unity3D para que tuviera las dimensiones necesarias.

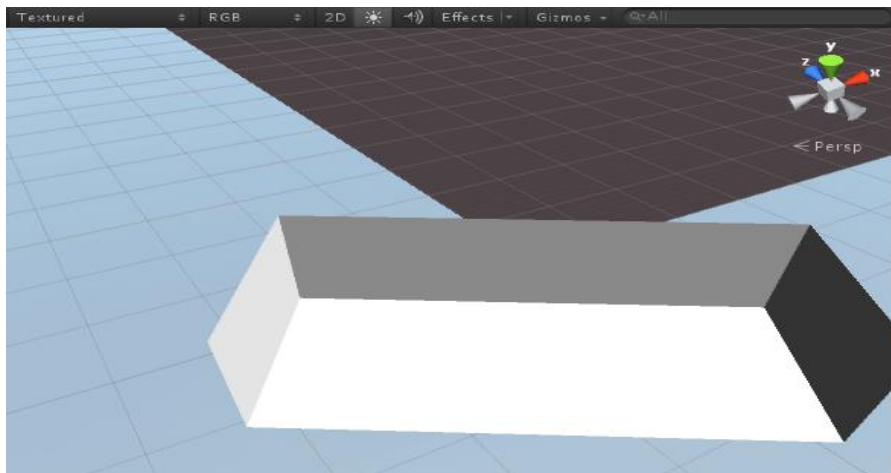


Figura 23: Modelo inicial del exterior del container de entrenamientos

Una vez escalado, rotado y situado en las coordenadas deseadas, se añadieron *Colliders* que gestionaran las colisiones con los demás elementos y otorgaran al container el comportamiento de un elemento rígido. Del mismo modo se añadieron cubos adicionales que permitiesen separar físicamente los distintos entrenamientos.

La dificultad de los ejercicios puede hacer que el usuario pueda quedar atascado en algún punto del escenario. Por ejemplo puede quedar bloqueado entre dos muebles sin saber cómo salir. Para evitar estas situaciones se ha añadido la funcionalidad llamada “respawn”. Esta funcionalidad consiste en añadir en cada escenario una posición de seguridad, libre de obstáculos, a la que pueda volver el usuario con sólo pulsar un botón. Desde esta posición puede volver a intentar el ejercicio sin la necesidad de empezar el juego desde

el inicio. Además se añadieron tres cámaras de presentación y tres luces para presentar los ejercicios en la selección de entrenamientos. El resultado final del container se puede observar en la figura 24.

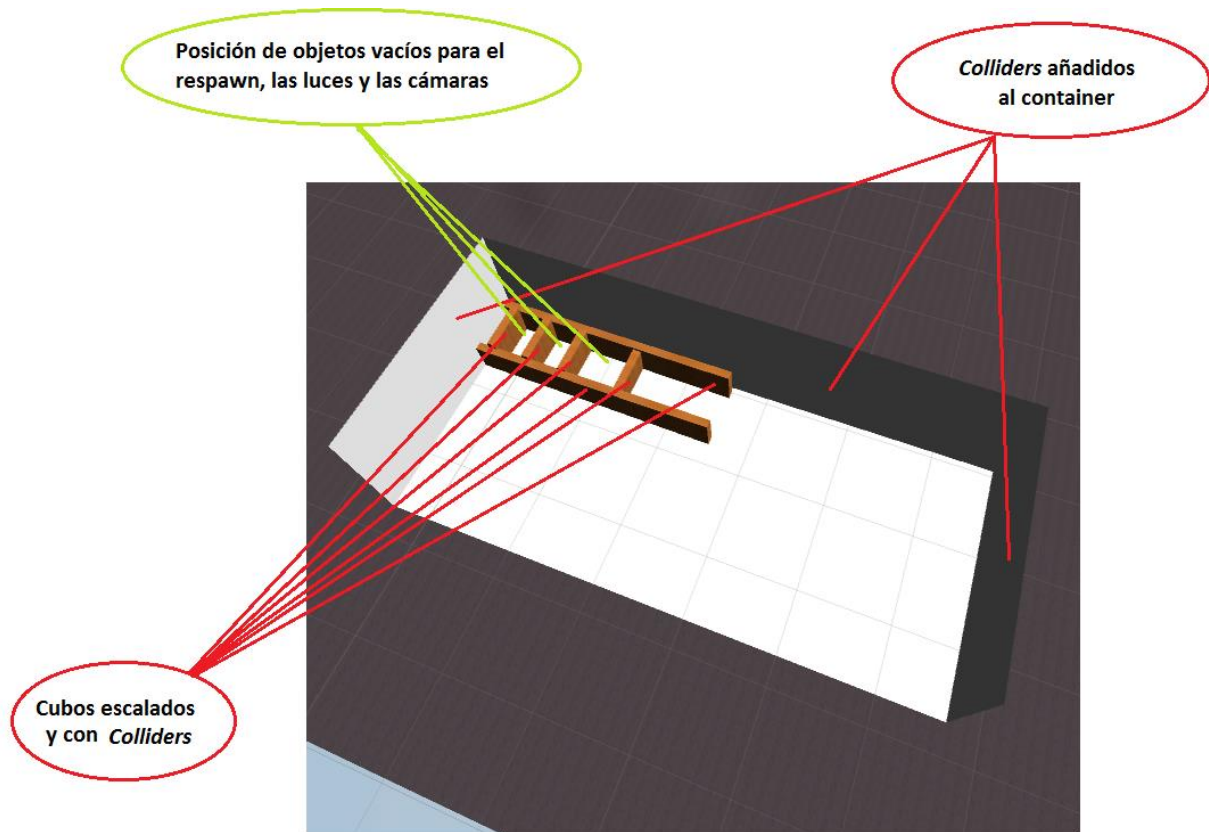


Figura 24: Modelo final del container de entrenamientos

3.3.2. Escenarios y componentes

Dentro del container se encuentran los tres escenarios, cada uno correspondiente a un ejercicio.

Ejercicio 1 – Habitación

El modelo de la habitación empleada en el primer ejercicio se ha descargado del repositorio de Google SketchUp Gallery. Estos modelos inicialmente no contaban con ningún *Collider* que les otorgara rigidez y delimitara el espacio en el que podría moverse la silla de ruedas y fue

necesario añadirlos para evitar por ejemplo que la silla pudiera atravesar las paredes y el usuario pudiera experimentar las diferentes colisiones con estos elementos del entorno.

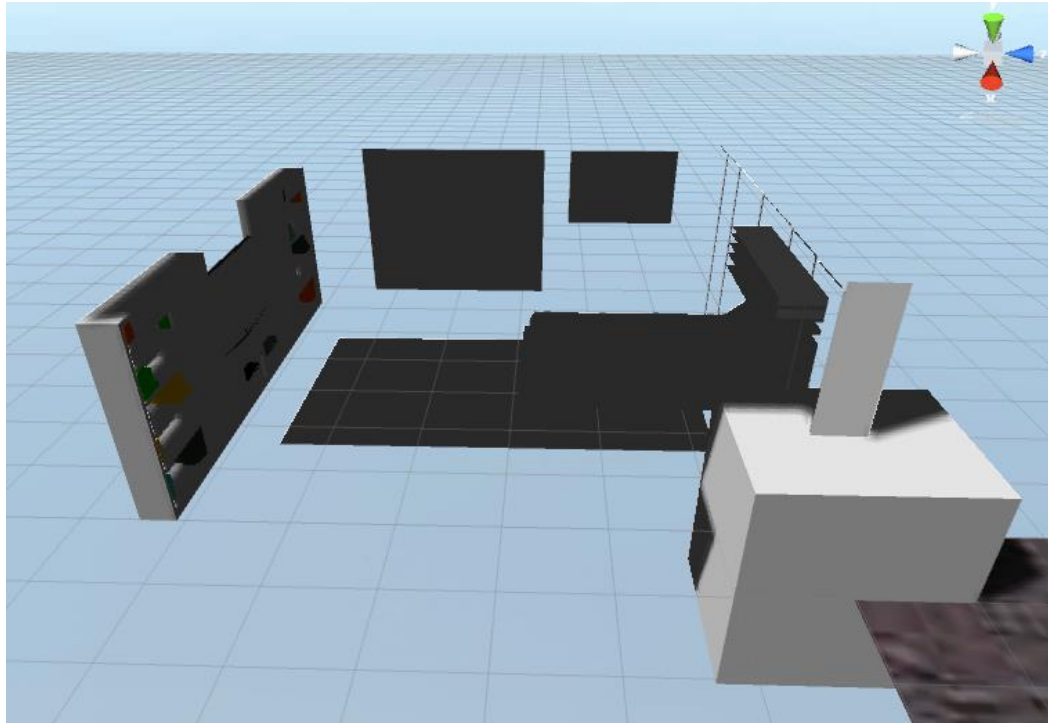


Figura 25: Modelo inicial del ejercicio 1 (Habitación)

Una vez importado el modelo en Unity3D, se escaló, se rotó y se situó en el interior del container contenedor. Una vez posicionado, se le añadieron los colisionadores para simular las paredes a los ejercicios, se eliminó el cubo de al lado de la cama (a la que también se añadió un Collider) ya que dificultaba la visión global de la habitación. Además se introdujo una mesita de noche al lado de la cama (obtenida del mismo modo que el modelo principal) para que otorgara cierta dificultad a la colocación de la silla de ruedas en el primer punto de control.

Al modelo generado se le añadieron también otros objetos que proporciona por defecto Unity3D: una esfera que anclaría a la silla de ruedas en paralelo a la cama y que cambiaría de color cuando la silla se colocara en su posición, tres cilindros que se situarían encima de la cama, cada uno

correspondiente a una de las etapas que deberá seguir el usuario para subirse a la cama, y una cámara ,que sustituiría en este caso a la de tercera persona, con un ángulo de visión que sirviera de ayuda en caso de encontrar tediosa la colocación de la silla en primera persona.

Para finalizar y una vez probado el ejercicio, se reposicionó y se escaló (escala menor) la cama para facilitar la ejecución del mismo y se le añadió un muro con texto (para su presentación). En la figura 26 se puede observar el resultado final del modelo de la habitación.

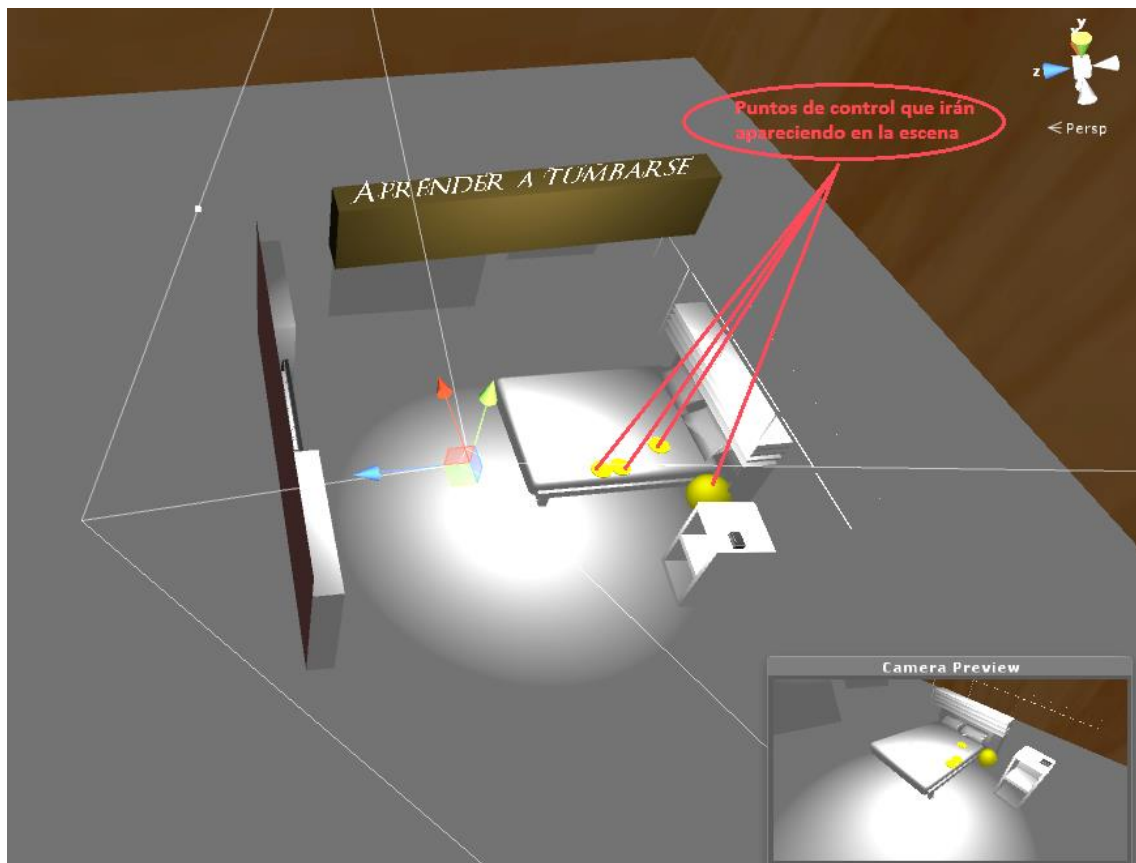


Figura 26: Modelo final del ejercicio 1 (Habitación)

Ejercicio 2 – Cocina

El modelo de la cocina empleado para la simulación del segundo ejercicio también se ha obtenido de [21]. Las características del modelo una

vez importado tampoco disponían de colisionadores y aparecía igualmente un cubo como consecuencia de haber importado el modelo en formato .dae previamente en Blender (éste lo introduce por defecto), para exportarlo a .fbx, compatible con Unity3D.



Figura 27: Modelo inicial del ejercicio 2 (Cocina)

Tras importar el modelo en el entorno de desarrollo, se comenzó por eliminar el cubo. Una vez obtenido el modelo deseado, se escaló, rotó y posicionó en el interior del recinto contenedor.

El siguiente paso fue añadir los colisionadores que otorgarían realismo a las colisiones con las distintas partes de la cocina y el muro con texto para la presentación del ejercicio.

Para finalizar se introdujeron los componentes con los que el usuario realizaría el ejercicio: un objeto vacío que simulara el recipiente del fregadero y permitiera detectar sus colisiones y dos modelos que el usuario tendría que

recoger de la encimera de la cocina, una sartén y un vaso (ambos editados usando el editor de Unity3D y descargados de [21]).

El resultado del modelo del segundo se ilustra en la figura 28:



Figura 28: Modelo final del ejercicio 2 (Cocina)

Ejercicio 3 – Salón

El modelo del salón utilizado se ha obtenido de [21]. Al igual que en los casos anteriores fue necesario añadir a mano el sistema de colisiones.

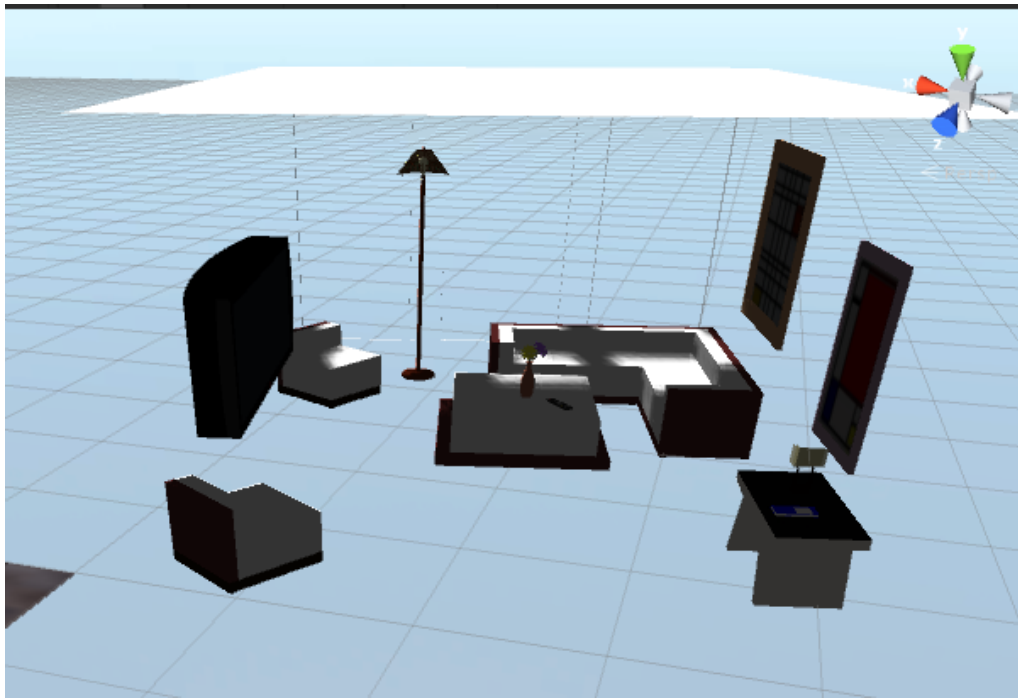


Figura 29: Modelo inicial del ejercicio 3 (Salón)

Una vez situado en el interior del container contenedor, se le añadió un muro con texto, un cubo que actuara de punto de control y uno elementos curvados (también descargados de [21]) que guiarán al usuario para situar la silla al comienzo del ejercicio.

Para finalizar se incorporaron los componentes con los que interactuarán los usuarios (en este caso al contar con varios elementos de decoración como sofás y mesas, la tarea fue mucho más compleja), una serie de bolas que cambiaran de color, en función de la colisión o no de las manos del personaje virtual asociado a la monitorización, a modo de puntos de control.

El resultado final del modelo del tercer y último ejercicio puede observarse en la figura 30.



Figura 30: Modelo final del ejercicio 3 (Salón)

3.3.3. Personajes

En el menú de selección, el sistema otorga la posibilidad de escoger entre un personaje femenino y otro masculino. Sin embargo, esta elección es puramente visual, ya que ambos tienen prácticamente el mismo comportamiento.



Figura 31: Selección de género de personaje

En la versión inicial del simulador [3] los personajes contaban con un componente “animator” que hacían que el personaje tuviera un aspecto más realista al realizar movimientos espontáneos como por ejemplo mirar la hora (ver figura 30) o mover las manos. En esta nueva versión del simulador fue necesario eliminarlo para no interferir con el movimiento real derivado de la monitorización del usuario con el dispositivo de detección de movimiento Xtion. Este componente fue sustituido por el script de control “ZigSkeleton.cs” que transfiere la información de movimiento al avatar. Por último fue necesario posicionar las partes del cuerpo de forma que una vez colocados, se ajustaran a las dimensiones de la silla.

Personaje masculino

El modelo de este personaje se obtuvo del plugin de Zigfu. Fue necesario añadirle cuatro colisionadores (en las dos manos y en los dos pies) que permitieran la implementación de la base lógica de los tres ejercicios de entrenamiento. En el caso de las manos, también disponen de dos ficheros de control, uno que permite al personaje simular el cogerse un pie y otro que es el núcleo de ejecución del tercer ejercicio (fortalecimiento de brazos).

En la figura 32 puede observarse la comparación entre el modelo inicial con el que se partía y el final con el cual interaccionará el usuario.



Figura 32: Modelo inicial hombre vs modelo final hombre

Personaje femenino

El modelo de este personaje se obtuvo del directorio “Prefabs” del proyecto de partida. Al igual que con el personaje masculino, se le añadieron cuatro colisionadores con las mismas características y elementos de control.

En este caso fue necesario eliminar los componentes “*Rigidbody*” y “*CharacterJoint*” que permitían simular el esqueleto del personaje, ya que el movimiento monitorizado de interés afectaba sólo a la parte superior del cuerpo, y estos elementos provocaban un movimiento en cadena del resto de las partes del cuerpo y hacían que el personaje no permaneciera anclado a la silla.

En la figura 33 puede observarse la comparación entre el modelo inicial con el que se partía y el final con el cual interaccionará el usuario.

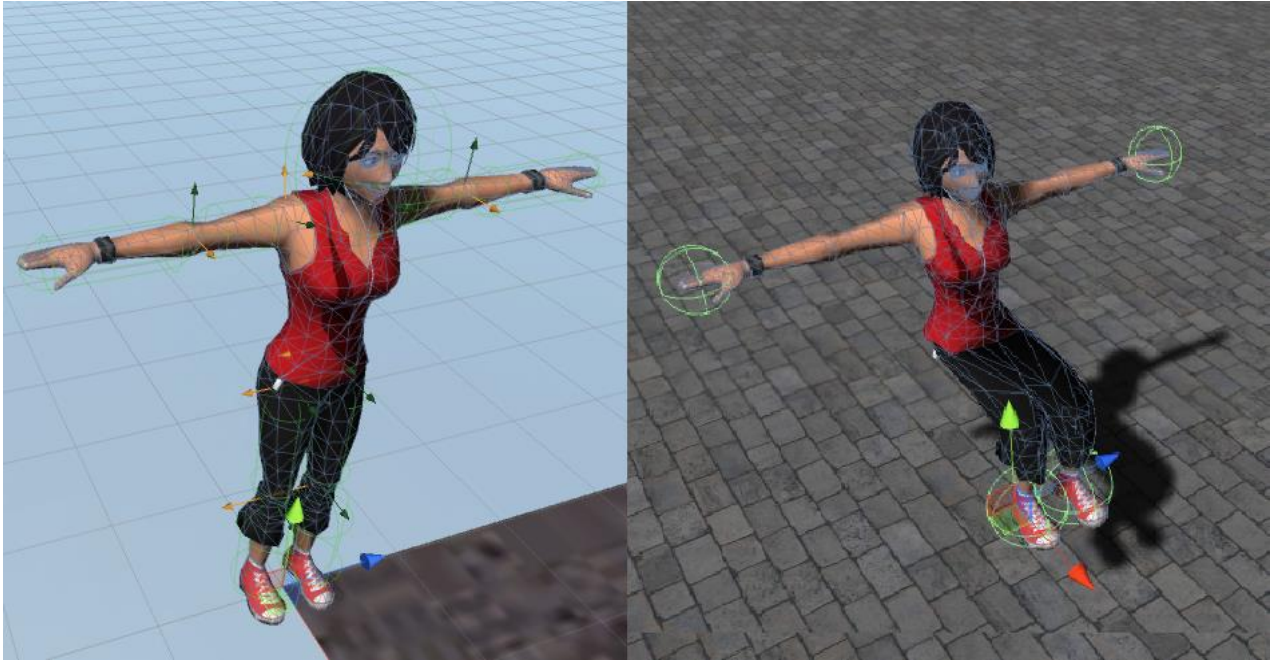


Figura 33: Modelo inicial mujer vs modelo final mujer

3.3.4. Puntos de control y audio

Como ya se ha detallado, los modelos de los ejercicios 1 y 3 tienen una serie de componentes que actúan como puntos de control por los que tiene que pasar el usuario hasta completarlos todos.

Nada más comenzar, se reproduce un audio que indica cual es la tarea a realizar y el modo correcto de hacerla. A medida que el usuario va llegando a los puntos de control, el audio reproducido cambia, para informar de cuál sería el siguiente paso a realizar y además el componente de la escena correspondiente cambia su color a verde para indicar visualmente la superación de la etapa.

Los nuevos componentes del ejercicio irán apareciendo (en color amarillo) a medida que se van completando los anteriores (que permanecerán activos y en color verde) una vez finaliza la reproducción del audio de ayuda correspondiente.

En la figura 34 se muestra un ejemplo del resultado de ir completando el ejercicio 3 y como van apareciendo y cambiando de color los componentes que actúan de puntos de control.

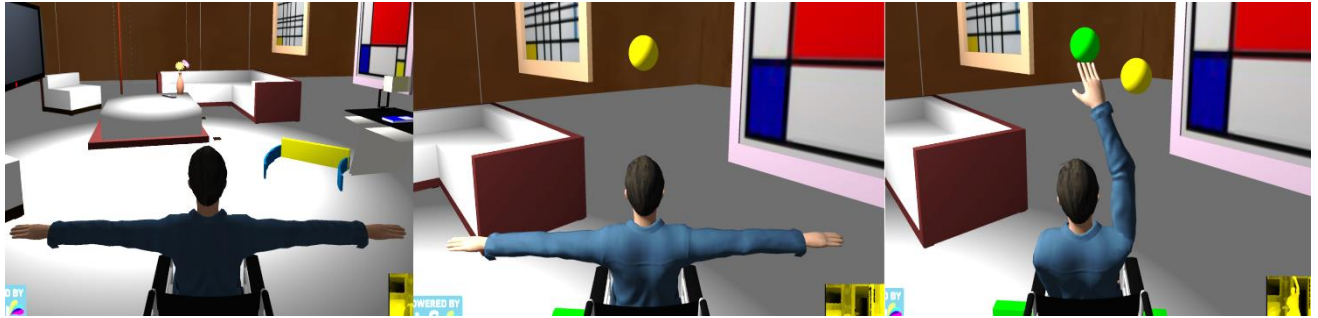


Figura 34: Transición entre puntos de control (ej: ejercicio 3)

3.3.5. Gestión de cámaras y navegación

Además de indicar al usuario el orden que tiene que seguir para realizar el ejercicio, el sistema dispone de un mecanismo de gestión de cámaras que se ejecuta en el fichero principal de control del modo de entrenamientos, *“ManageExercises.cs”*. Este script permite cambiar entre visión en primera y tercera persona (excepto en el ejercicio 1, que se decidió sustituir la tercera persona por una visión desde un punto más lejano (en una esquina del techo) para facilitar al usuario la visión global y por tanto la colocación de la silla de ruedas).

Las diferentes opciones de cámara también incorporan una flecha a modo de brújula, que cambiará de sentido una vez se supere un punto de control, de forma que siempre apunte hacia el próximo destino al que dirigimos.

En la figura 35 se puede observar las diferentes opciones de las que dispone el usuario para visualizar el escenario:

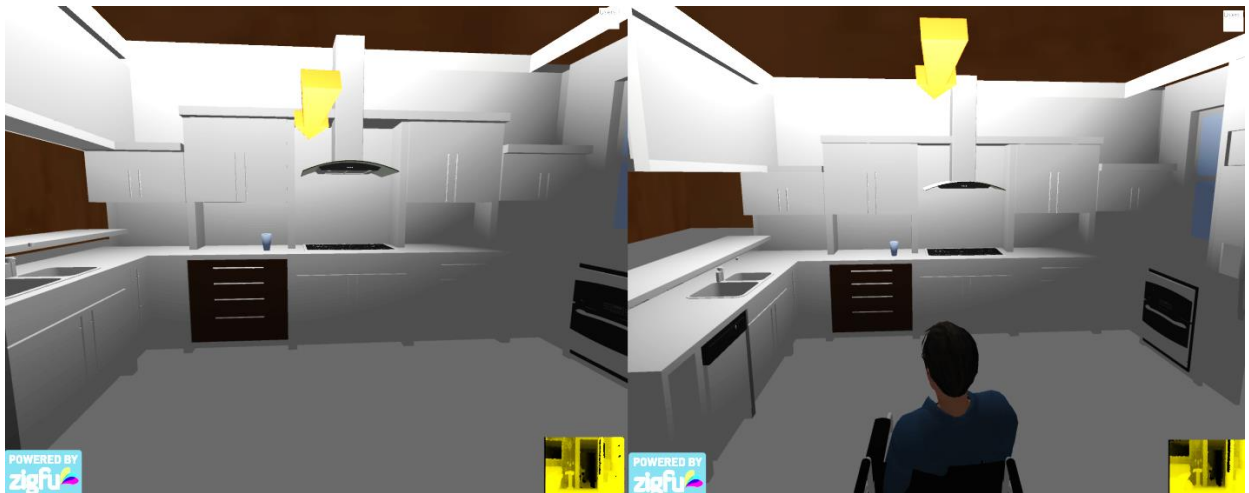


Figura 35: Vista primera persona vs vista tercera persona

Además como funcionalidad adicional a la cámara de primera persona, se le añadió un control de giro de visión que permitiera aumentar el ángulo de visión del usuario y que tratara de simular el movimiento de giro del cuello.

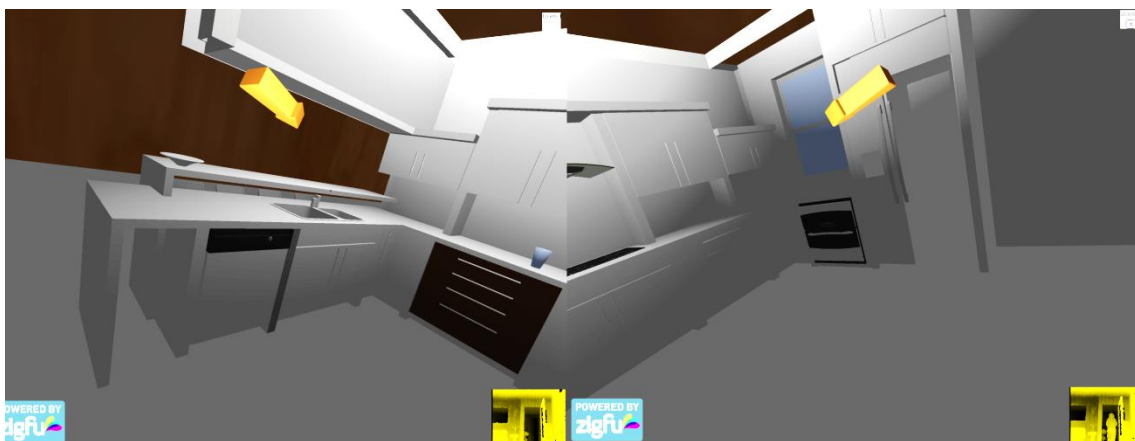


Figura 36: Vista girada a la izquierda vs vista girada a la derecha

Se pensó en anclar la cámara de primera persona al personaje de forma que el giro del cuello se reflejara en el ángulo de visión del escenario, sin embargo esto provocaría una vibración debida a la imprecisión del sistema de tracking en la cabeza del avatar que resultaba molesto para el usuario. Por ello,

finalmente se optó por desechar esta opción, aunque desde el punto de vista técnico no sería un problema incluirla si se dispone de sistemas de monitorización más precisos.

3.4. Estructura del simulador

El simulador que se ha desarrollado comprende dos apartados claramente diferenciados: la parte hardware y la parte software del sistema.

El nivel de hardware comprende toda la parte física del simulador y está compuesta por el ordenador, la plataforma de movimientos, los dispositivos HMD, el Joystick y el dispositivo de detección de movimiento.

El nivel software hace referencia al soporte lógico del sistema, e incluye aspectos como el diseño del menú, las físicas asociadas al movimiento de la silla o los personajes, y los cambios en el escenario asociados a los resultados del usuario en el ejercicio.

3.4.1. Comunicación hardware

El sistema se ejecutará en un ordenador, que actuará de núcleo o nodo central y gestionará la comunicación con cada uno de los dispositivos que se han utilizado en este proyecto. A continuación se resumen las principales comunicaciones que se han establecido (especificando además el sentido de las mismas):

- **Plataforma de movimiento:** el ordenador enviará una serie de órdenes y peticiones a la plataforma y ésta, responderá transmitiendo la información de su estado. La comunicación en este caso será bidireccional (CPU \leftrightarrow SMK3DOF).

- **Dispositivos HMD:** el ordenador será el encargado de transmitir en forma de datos la señal de video al display de las gafas. La comunicación en este caso será unidireccional (CPU → HMD).
- **Joystick:** este dispositivo permite transmitir la información de entrada del usuario que lo esté controlando al ordenador. En este caso la comunicación será unidireccional (Joystick → CPU).
- **Asus XtionPRO LIVE:** el ordenador enviará una serie de peticiones al dispositivo y este responderá transmitiendo la información de su estado. En este caso la comunicación será bidireccional (Xtion ↔ CPU).

En resumen, en la figura 37 se podrá observar de una forma sencilla cual es el esquema de intercambio de información entre los distintos dispositivos empleados:



Figura 37: Esquema de comunicaciones a nivel hardware

3.4.2. Diseño y estructura software

El proyecto se ha estructurado en una serie de directorios que se ha nombrado en función del contenido almacenado y la tarea que este lleva a cabo en el sistema. A su vez, todo el simulador se almacena en un directorio principal, identificado como Assets (nombre por defecto de Unity).

En este proyecto se han añadido los directorios “ZigFu” y “OVR”, con las Apis necesarias para controlar Xtion y las gafas Oculus Rift respectivamente. Además se han incluido en el directorio “Scripts” todos los scripts utilizados en el desarrollo de la monitorización. “Prefabs” incluye todos los agrupamientos de objetos y funcionalidades en una misma unidad, usados como por ejemplo en el recinto interior en las escenas. En “modelos” se incluyen aquellos modelos 3D utilizados para componer las escenas y en “sonidos” los sonidos empleados para explicar y guiar la ejecución de los ejercicios.

En la figura 38 se muestra el desglose de los directorios que componen la estructura Software, los directorios nuevos se muestran recuadrados en verde y los que han sido modificadas respecto a la versión anterior del simulador en rojo:

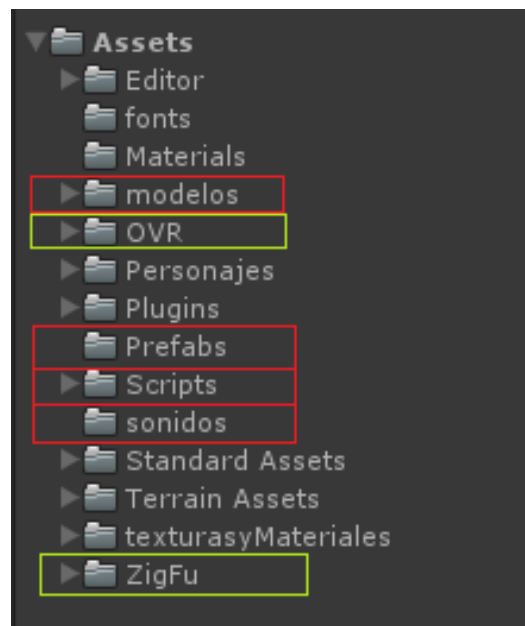


Figura 38: Estructura Software en directorios

El directorio “Zigfu” incluye ficheros que permiten el reconocimiento del dispositivo de detección de movimiento y se han podido importar en Unity3D gracias al plugin OpenSource ZDK proporcionado por Zigfu. El contenido de este directorio es totalmente de Zigfu, ya que los scripts que han sido modificados se han trasladado al directorio correspondiente (“DispositivoXtion”) dentro del directorio “Scripts/ComunicacionXtion”.

En el directorio “OVR” se encuentran todos los recursos (scripts, prefabs, texturas, etc.) que son necesarios para la correcta comunicación entre las Oculus y el entorno de Unity3D y que se descargaron de la página web de Oculus VR (<http://www.oculusvr.com/>). Como trabajo adicional fue necesario implementar dos scripts que controlaran automáticamente el manejo de cámaras en función de la disponibilidad o no de las mismas.

El contenido del directorio “Scripts” compone la parte fundamental del desarrollo. Su contenido se ha modularizado en directorios independientes en función de la funcionalidad proporcionada:

- **“Comunicacion3DOF”**: directorio que contiene los scripts necesarios para controlar la comunicación con la plataforma de movimiento y para gestionar las escenas de movimiento guiado y libre.
- **“ComunicacionXtion”**: directorio que contiene los scripts necesarios para controlar la comunicación con el dispositivo de detección de movimiento y gestionar los ejercicios de entrenamiento.

Para analizar en más detalle la estructuración de los scripts involucrados en el desarrollo de este proyecto, en la figura 39 se muestra un desglose del directorio “Scripts”, indicando que archivos se han añadido nuevos (en verde) y cuales se han modificado (en rojo).

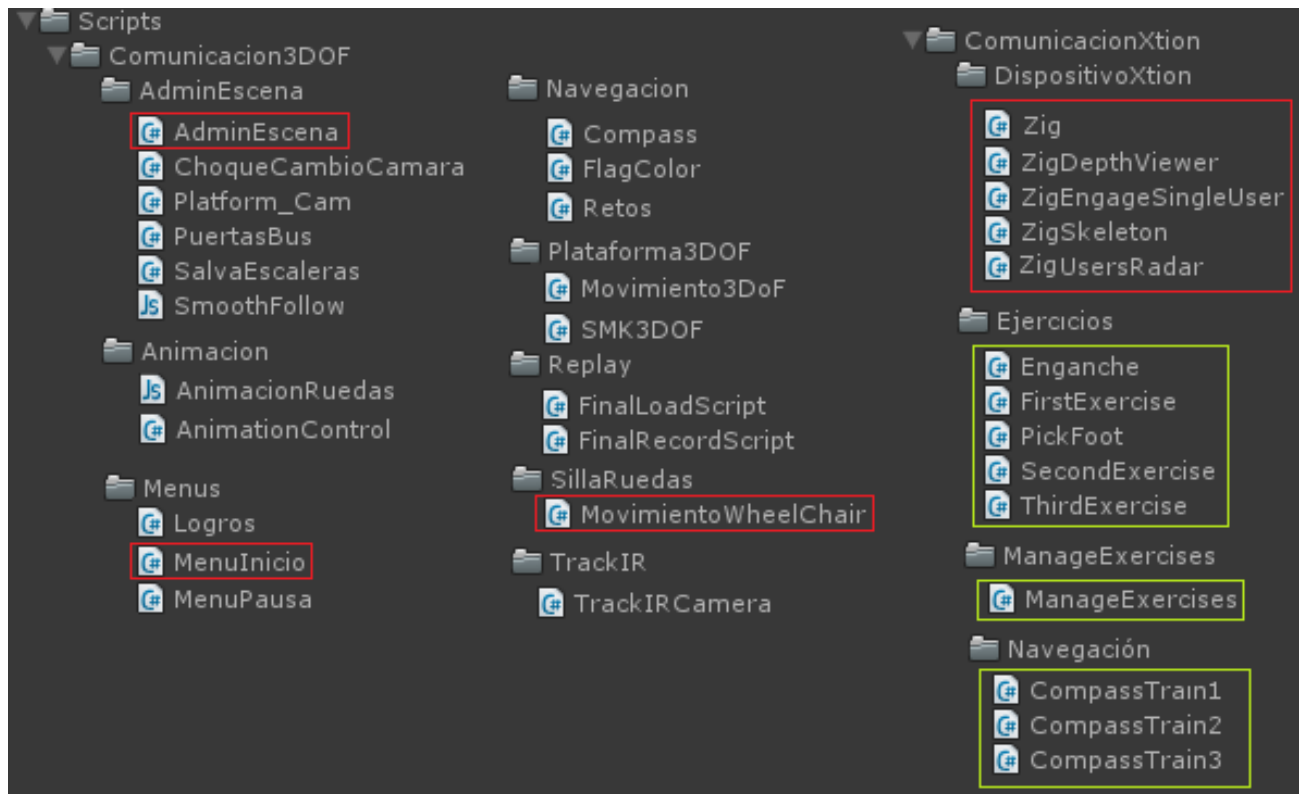


Figura 39: Desglose del directorio “Scripts”

A su vez, cada uno de los módulos principales del directorio “Scripts”, “Comunicacion3DOF” y “ComunicacionXtion”, pueden dividirse en dos subsistemas:

- **El sistema de comunicación:** contiene aquellos scripts que permiten el reconocimiento y la sincronización del dispositivo empleado dentro del motor de desarrollo de Unity3D.
- **El sistema de control de escenas:** contiene aquellos scripts que componen la lógica de los escenarios, de sus componentes y además gestionan los datos recibidos por parte del sistema de comunicación.

En el siguiente esquema se muestra la visión global de proyecto y la relación entre directorios en función del sistema al que pertenecen.

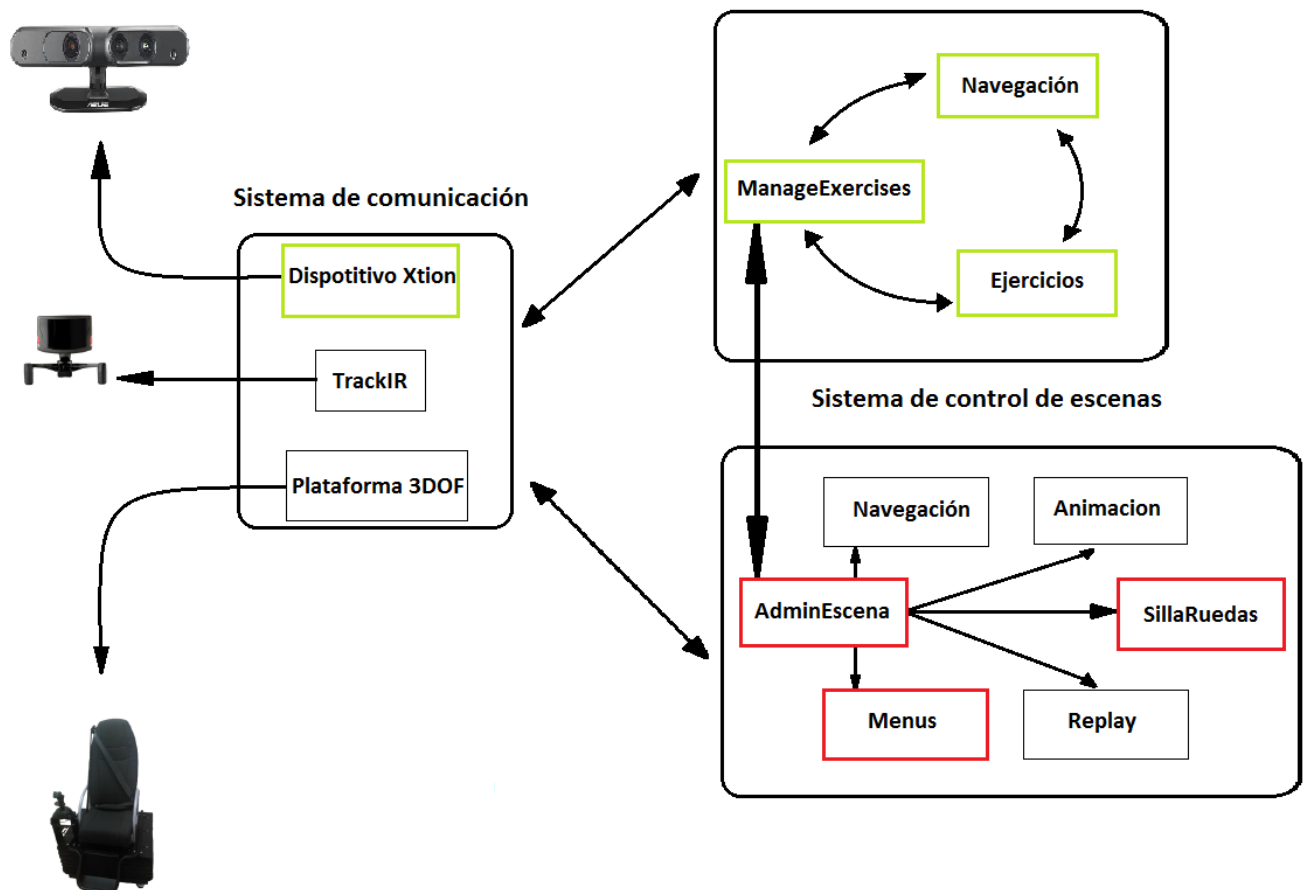


Figura 40: Esquema de los sistemas que componen el Software del simulador

A continuación se describen con detalle los directorios y los scripts añadidos o modificados, que conforman cada sistema del simulador:

❖ Sistema de comunicación:

○ Directorio 1: Dispositivo Xtion

Contiene aquellos scripts (programados en C#) encargados de gestionar la integración del dispositivo de detección de movimiento en el entorno de desarrollo de Unity3D, ya que el dispositivo XtionPRO no dispone de un plugin específico para trabajar en este programa. Todos los scripts se han obtenido del plugin de Zigfu para Unity3D:

- **“Zig.cs”**: se trata del fichero principal, que permite asociar un escuchador al modelo que se quiera asociar a la monitorización del usuario. Además, gestiona el tipo de entrada, la configuración del dispositivo a conectar (ya que también es compatible con Kinect), y el estado del usuario monitorizado.

El código de la figura 42 muestra a modo de ejemplo como se crea un objeto asociado a un escuchador.

```
for (int i = 0; i < listeners.Count; ) {  
    GameObject go = listeners[i];  
    if (go) {  
        go.SendMessage(msgname, arg, SendMessageOptions.DontRequireReceiver);  
        i++;  
    }  
    else {  
        listeners.RemoveAt(i);  
    }  
}
```

Figura 41: Gestión de escuchadores de Zig.cs

- **“ZigDepthViewer.cs”**: encargado de crear un mapa de resolución variable en función del dispositivo.
- **“ZigEngageSingleUser.cs”**: hace uso de escuchadores y mensajes “UserEngaged” y “UserDisengaged”, para captar al usuario a monitorizar.

En la figura 42 se muestra a modo de ejemplo como se añade un escuchador al usuario que se está monitorizando.

```

void EngageUser(ZigTrackedUser user) {
    if (null == engagedTrackedUser) {
        engagedTrackedUser = user;
        foreach (GameObject go in EngagedUsers) user.AddListener(go);
        SendMessage("UserEngaged", this, SendMessageOptions.DontRequireReceiver);
    }
}

```

Figura 42: Engaged de usuario

- “*ZigUsersRadar.cs*”: se encarga de crear un recuadro en la pantalla con el número de usuario captado que permite saber al usuario si se ha iniciado o no la monitorización.
- “*ZigSkeleton.cs*”: permite reconocer puntos clave durante la monitorización (cabeza, manos, brazos, etc.) y extraer la información de su posición y rotación.

❖ Sistema de control de escenas:

○ Directorio 1: AdminEscena (**MODIFICADA**)

En este módulo el único script (programado en C#) que se ha modificado es el “AdminEscena.cs”. Los cambios realizados consisten en añadir una constantes globales del modo de juego, para que no se solapen las ejecuciones de los modos libre/guiado y entrenamientos.

○ Directorio 2: Menus (**MODIFICADA**)

Las principales modificaciones se han realizado sobre el script (programado en C#) “MenuInicio.cs”, para añadir al menú la opción asociada a la nueva funcionalidad:

- Añadir a la escena el modelo del texto de la nueva opción.
 - Modificar la lógica de selección (se pasa de dos a tres opciones).
 - Gestionar la variables de “AdminEscena.cs” y del fichero de control de ejercicios que se verá más adelante, “ManageExercises.cs”.
 - Gestión de activación/desactivación de modelos según la modalidad, agilizando la ejecución global del simulador.
- **Directorio3: SillaRuedas (MODIFICADA)**
- Contiene un único script, “MovimientoWheelChair.cs” (programado en C#), que se ha modificado para que la nueva modalidad limite la velocidad, el par motor y la fuerza de frenado de la silla.

```
if (ManageExercises.estado==ManageExercises.JUGANDO){
    MaxVelocity = 2.2f;
    EngineTorque = 95f;
    BreakForce=15f;
}
```

Figura 43: Modificación de las características de la silla de ruedas

- **Directorio4: ManageExercises (CREADA)**
- Directorio que contiene el script (programado en C#) principal (“ManageExercises.cs”) que gestiona la modalidad de entrenamientos y controla sus parámetros generales: configuración inicial, selección de ejercicios, gestión de cámaras y giros, respawning¹⁹ y finalización.

¹⁹ Funcionalidad que permite, cuando el usuario se queda atascado en la escena, volver a una posición segura cercana al punto donde se produjo el atasco.

- **Directorio5: Ejercicios (CREADA)**

Este directorio se compone de seis scripts (programados en C#), que son los encargados de gestionar la lógica de los ejercicios:

- *“FirstExercise.cs”*: encargado de gestionar las colisiones de las partes del cuerpo del personaje con los puntos de control del primer ejercicio (transición silla de ruedas – cama).
- *“PickFoot”*: script auxiliar al primer ejercicio para la gestión de las jerarquías de los elementos en colisión.
- *“SecondExercise.cs”*: encargado de gestionar las colisiones y las jerarquías de los elementos involucrados en el segundo ejercicio.
- *“ThirdExercise.cs”*: encargado de gestionar las colisiones de los elementos involucrados en el tercer ejercicio.
- *“Enganche.cs”*: script auxiliar para la colocación de la silla en la posición adecuada que permita la correcta ejecución del tercer ejercicio.

- **Directorio6: Navegacion (CREADA)**

Este directorio se compone de tres scripts (programados en C#) con la misma funcionalidad (cada uno referente a un ejercicio):

- *“CompassTrain1.cs”*, *“CompassTrain2.cs”*,
“CompassTrain1.cs”: scripts encargados de la gestión de los puntos de control del ejercicio, del giro del indicador del siguiente paso a realizar y de las corutinas de reproducción de audio del ejercicio correspondiente.

3.5. Integración hardware

3.5.1. Dispositivo de detección de movimiento



Figura 44: Asus XtionPRO LIVE integrado en el proyecto

Se trata del elemento hardware en torno al cual se ha desarrollado la ampliación del simulador. Consiste en un dispositivo multifunción, en el que todas las funcionalidades están orientadas a la detección:

- Detección de gestos.
- Detección de cuerpo completo, facilitando el reconocimiento de múltiples usuarios.
- Detección de colores RGB.
- Detección de voz.

Debido a sus escasas dimensiones (18 x 3,5 x 5 cm) y su ligero peso (165 g) es un elemento fácilmente portable cuyo uso se recomienda en espacios interiores. Cuenta con un sensor de profundidad y otro de RGB, con

un campo de visión de 58° Horizontal, 45° Vertical y 70° Diagonal y una distancia de uso de entre 0.8 y 3.5 m.[20]

Se conecta al PC a través de un puerto USB 2.0 y es necesario instalar un paquete de software con los controladores necesarios para que se reconozca como dispositivo externo:

- OpenNI Framework (Versión 1.2.0.7).
- Sensor DDK (Versión 5.0.2.3).
- NITE (Versión 1.3.1.6).
- USB driver (Versión 3.1.2.0).

Interacciona con el entorno de desarrollo de Unity3D haciendo uso de los scripts de control proporcionados por Zigfu en el plugin ZDK_Unity40_1.1_trial.unitypackage, desarrollados en C#, aunque también soporta C++. Es compatible con los principales sistemas operativos (Linux Ubuntu y Windows XP y 7).

3.5.2. Dispositivos HMD

Para incrementar el nivel de inmersión en una realidad virtual, se han utilizado dos dispositivos HMD (Head-Mounted Displays). Esta clase de dispositivos permiten la visualización a modo de casco de imágenes creadas por un ordenador sobre un display cercano al ojo.

La principal diferencia entre ellos es que las Vuzix se han utilizado en modo 2D, es decir, los dos ojos ven exactamente la misma imagen, mientras que en las Oculus Rift el usuario experimenta una verdadera sensación de 3D, ya que recibe una imagen diferente en cada ojo.

Con respecto a la usabilidad, las primeras pruebas han demostrado que la mayoría de los usuarios experimentan una sensación de inmersión con las Vuzix y no se dan cuenta de que no están viendo en 3D hasta que no prueban la segunda opción (Oculus Rift).

Sin embargo, aunque la versión 3D ofrece una imagen mucho más realista del entorno, la mayoría de los usuarios manifestaron sentirse mareados a los pocos segundos de utilizarlas, por lo que parece menos usable para realizar los entrenamientos.

Vuzix iWear VR920



Figura 45: Vuzix iWear VR920 integrada en el proyecto

Elemento hardware que permite al usuario ver una imagen sin la necesidad de fijar la mirada en la pantalla del ordenador. Las características principales de este dispositivo son:

- Conector USB para la alimentación y VGA para la transmisión de imagen y audio.
- iWear 3D que permite control 2D y 3D automático.
- Micrófono y auriculares incorporados.
- Head-tracker 3DOF integrado.
- Display con frecuencia de 60 Hz

Aunque las gafas Vuzix permiten visión 3D enviando frames alternos con distinta frecuencia a cada ojo, no es una solución trivial y su desarrollo escapa al alcance de este proyecto.

Oculus Rift VR



Figura 46: Oculus Rift VR integrada en el proyecto

Se trata de un dispositivo de realidad virtual con baja latencia, desarrollado por Oculus VR²⁰ y orientado para desarrolladores. La versión utilizada en este proyecto es la 1 (Development Kit 1) y sus principales características son:

- Pantalla de 7 pulgadas.
- Pantalla de LCD y profundidad de 24 bits por píxel.
- 3D estereoscópico sin superposición 100% (el ojo izquierdo ve área adicional a la izquierda y el ojo derecho ve área adicional a la derecha) que imita a la visión del ser humano.

²⁰ <http://www.oculusvr.com/>

- Campo de visión de 90 grados horizontal y 110 diagonal.
- Resolución 1280 × 800 (con relación de aspecto 16:10), que conduce a un efectivo de 640 × 800 por ojo (con relación de aspecto 4:5).
- Interfaz USB para enviar datos de seguimiento a la máquina servidor y encender el dispositivo y entradas DVI y HDMI para la transmisión de la imagen.

Una de las principales restricciones con las que cuenta este dispositivo respecto al anterior es la ausencia de auriculares para reproducción de audio.

En este proyecto se ha utilizado como display en 3D que permite al usuario sumergirse en una visión virtual muy parecida a la realidad.

3.5.3. Logitech Dual Action



Figura 47: Logitech Dual Action integrado en el proyecto

Este dispositivo actúa como controlador del vehículo (en sincronización con el teclado). Creado por la empresa Logitech²¹, dispone de dos mandos, cuatro flechas de selección y 10 botones, y su instalación únicamente requiere el conectarse a través de un puerto USB al ordenador. El entorno de desarrollo de Unity3D lo reconoce automáticamente y permite detectar la pulsación de botones u orientación de los mandos del Joystick.

A screenshot of a code editor showing the C# method `Input.GetAxisRaw("Vertical")`. The text is highlighted in blue, and the entire line is enclosed in a black rectangular box with a slight drop shadow.

Figura 48: Detección de movimiento vertical de Joystick

A screenshot of a code editor showing the C# method `Input.GetKeyDown(KeyCode.Joystick1Button1)`. The text is highlighted in blue, and the entire line is enclosed in a black rectangular box with a slight drop shadow.

Figura 49: Detección de pulsación de botón1

Se ha utilizado el mando izquierdo del Joystick para controlar el movimiento de la silla de ruedas y la selección de modos de juego y los botones para el cambio y giro de cámara y la confirmación de ejercicio de entrenamiento a realizar. Esto permite al usuario experimentar una simulación más atractiva gracias a la similitud con el control de dirección que tendría un discapacitado en una silla de ruedas motorizada.

3.6. Requisitos del simulador

La extensión del simulador de partida para integrar un dispositivo de detección de movimiento que permitiera la monitorización del usuario requería

²¹ <http://www.logitech.com/es-es>

de una serie de aspectos a cumplimentar, los cuales se pueden dividir por un lado en requisitos funcionales, y por otro, en requisitos no funcionales.

Los requisitos funcionales están relacionados con el modelado y desarrollo en Unity3D, la implementación de la lógica necesaria para la integración del dispositivo XtionPRO LIVE y su posterior utilización en un escenario, y la integración hardware

Los requisitos no funcionales están relacionados con los principios de diseño que guiarán todo el desarrollo.

3.6.1. Requisitos Funcionales

Tabla 3: Requisito funcional RF – 01

Identificador	RF – 01
Nombre	Sincronización Xtion – Personaje
Descripción	Reflejar en el movimiento del avatar del usuario el movimiento real que está siendo captado por el dispositivo Xtion.
Obligatoriedad	Obligatorio

Tabla 4: Requisito funcional RF – 02

Identificador	RF – 02
Nombre	Sincronización Personaje – Silla de ruedas

Descripción	Mantener durante la monitorización del usuario un movimiento solidario entre los dos elementos
Obligatoriedad	Obligatorio

Tabla 5: Requisito funcional RF – 03

Identificador	RF – 03
Nombre	Independización del escenario de partida
Descripción	Diseñar la modalidad de entrenamientos de forma que no interfieran con los modos libre y guiado
Obligatoriedad	Obligatorio

Tabla 6: Requisito funcional RF – 04

Identificador	RF – 04
Nombre	Ejercicios demostrativos
Descripción	Diseñar una serie de ejercicios que permitan al usuario realizar entrenamientos simulados
Obligatoriedad	Obligatorio

Tabla 7: Requisito funcional RF – 05

Identificador	RF – 05
Nombre	Ejercicio guiado
Descripción	Implementar un sistema que permita guiar al usuario en la realización de los ejercicios
Obligatoriedad	Obligatorio

Tabla 8: Requisito funcional RF – 06

Identificador	RF – 06
Nombre	Sistema de control de visión
Descripción	Permitir al usuario la visión en diferentes tipos de vistas (primera persona y tercera persona)
Obligatoriedad	Obligatorio

Tabla 9: Requisito funcional RF – 07

Identificador	RF – 07
Nombre	Control de giro de cámara

Descripción	Permitir el giro de la cámara durante la ejecución de los ejercicios de entrenamiento.
Obligatoriedad	Opcional

Tabla 10: Requisito funcional RF – 08

Identificador	RF – 08
Nombre	Asus XtionPRO LIVE
Descripción	Integración del dispositivo de detección de movimiento en el entorno de desarrollo de Unity3D
Obligatoriedad	Obligatorio

Tabla 11: Requisito funcional RF – 09

Identificador	RF – 09
Nombre	Oculus y Vuzix VR920
Descripción	Empleo de los dispositivos HMD a modo de display para visualizar la escena en 3D
Obligatoriedad	Obligatorio

Tabla 12: Requisito funcional RF – 10

Identificador	RF – 10
Nombre	Logitech Dual Action
Descripción	Control de los ejercicios de entrenamiento con este dispositivo externo como alternativa al teclado (con las mismas opciones de pulsación)
Obligatoriedad	Obligatorio

3.6.2. Requisitos No Funcionales

Tabla 13: Requisito no funcional RNF – 01

Identificador	RNF – 01
Nombre	Arquitectura modular y extensible
Descripción	La estructura interna de los ficheros implicados en la detección de movimiento deberá ser modular, permitiendo su integración en otras escenas o proyectos y añadir posibles funcionalidades extras con facilidad.
Obligatoriedad	Obligatorio

Tabla 14: Requisito no funcional RNF – 02

Identificador	RNF – 02
Nombre	Diseño sencillo e intuitivo
Descripción	El diseño del simulador deberá ser intuitivo tanto para el usuario final como para el desarrollador.
Obligatoriedad	Obligatorio

Para finalizar se incluye una matriz que referencia los objetivos (columnas) presentados en el apartado 1.4. Objetivos con los requisitos (filas), para representar que los objetivos definidos inicialmente quedan cubiertos por los requisitos.

Tabla 15: Matriz Objetivos vs Requisitos

	OF - 01	OF - 02	OF - 03	OF - 04	OT - 01	OT - 02	OT - 03	OT - 04	OT - 05
RF - 01	X					X			
RF - 02						X			
RF - 03		X					X		
RF - 04		X		X			X		
RF - 05									X
RF - 06			X					X	
RF - 07			X						
RF - 08	X				X				
RF - 09			X					X	
RF - 10							X	X	
RNF - 01			X	X			X		X
RNF - 02	X		X			X			X

Capítulo 4

Plan de validación

4.1. Introducción

El desarrollo se realizó en dos espacios distintos: uno era el laboratorio de la universidad que tenía muchos elementos que impedían disponer de un área vacía de aproximadamente 2x2 m y el otro era la residencia del desarrollador, donde se realizó la mayor parte del desarrollo debido a la necesidad de espacio para maniobrar (2x2 m) y que el rango de profundidad de captación de movimiento fuera el adecuado para la monitorización del usuario (1,5 m).

Durante el desarrollo se fueron realizando reuniones periódicas para evaluar el grado de desarrollo tanto de los requisitos funcionales como no funcionales y se realizaron pruebas sencillas con voluntarios. Durante el desarrollo del proyecto se realizaron también unas prácticas en la empresa Simumak, creadora de la plataforma de movimiento SMK3DOF. Esto ha permitido comprender mejor las etapas del proceso de desarrollo de un simulador y las pruebas (por ejemplo siempre es importante que los cambios subidos se testeen en una versión de prueba antes de subirla a la versión de validación).

Una vez realizada la integración de los elementos hardware y completado el diseño e implementación de las soluciones hardware, el siguiente paso a tomar es el de evaluar si la respuesta que nos ofrece la extensión del simulador que se ha desarrollado es la óptima.

4.2. Pruebas realizadas

El grueso de las pruebas referentes a la monitorización se han realizado en el domicilio del desarrollador, ya que se disponía tanto del dispositivo Xtion como del Joystick. Sin embargo, para las pruebas en las que se necesitaba la plataforma 3DOF y los dispositivos HMD, fue necesario el trabajo en el laboratorio.

A continuación, se las principales labores llevadas a cabo para obtener una monitorización lo más precisa posible:

- Ajuste de la posición relativa del dispositivo de detección de movimiento con respecto al usuario que pretende ser monitorizado. Es necesario que el usuario se sitúe a unos 1.5 m del dispositivo y que disponga de un área libre a su alrededor de unos 2x2 m.
- Asignación de los puntos de control necesarios para permitir reflejar el movimiento en el personaje de la forma más realista posible. Se fijaron como puntos de control la cabeza, el cuello, el torso, los hombros, los codos y las manos.
- Ajuste de los parámetros de amortiguamiento y de rotación de las partes del cuerpo monitorizadas. Se trabaja con un amortiguamiento de 30 grados para el ajuste de las rotaciones y de 15 grados para el ajuste de las posiciones.
- Ajuste de la posición de la cámara en primera persona para que no solape con el cuerpo del personaje en situaciones en las que el

usuario inclina su cuerpo hacia delante. Únicamente se necesitó adelantar la cámara 5 puntos en la dirección paralela al avatar.

Con respecto a los dispositivos HMD, se ha comprobado únicamente que su funcionamiento fuera el correcto.

4.3. Recursos utilizados para las pruebas

Todas las pruebas del sistema han sido realizadas en un mismo equipo, cuyas principales características se especifican en la siguiente tabla (tabla 13):

Tabla 16: Características del equipo de testeo

Marca	Acer ²²
Modelo	Aspire 5750G
Sistema operativo	Windows 7 Home Premium (64 bits)
Procesador	Intel® Core™ i7-2630QM CPU @2.00 GHz 2.00 GHz
Memoria RAM	8,00 GB (7,85 GB utilizable)
Tarjeta gráfica	Nvidia GT540M

²² <http://www.acer.es/ac/es/ES/content/home>

4.4. Requisitos del sistema

Dado que el elemento que más recursos consume es el motor de simulación que se ejecuta sobre Unity3D, se han utilizado los requisitos descritos en la página web de contacto de Unity3D [22] como requisitos mínimos del sistema para permitir la correcta ejecución del simulador (ya que se ha comprobado que estos requisitos siguen siendo suficientes al incluir los demás elementos):

- Windows XP con SP2 o posterior; Windows 7 con SP1 o posterior; Windows 8; Mac OS X Snow Leopard 10.6 o posterior. Tome en cuenta que Unity no se probó en las versiones de servidor de Windows y OS X. No soporta Windows Vista.
- Tarjeta gráfica con capacidad para DirectX 9 (shader modelo 2.0). Cualquier tarjeta fabricada desde 2004 debería funcionar.
- El uso de Occlusion Culling requiere una GPU con soporte Occlusion Query (algunos GPU con Intel no soportan eso).

Capítulo 5

Planificación del proyecto

El proyecto se ha dividido en cuatro etapas.

5.1. Ciclo de vida

- **Fase de inicio:** en esta etapa el principal objetivo ha sido familiarizarse con el entorno de desarrollo y los recursos hardware y software descritos en el apartado 3.4. Estructura del simulador, que se iban a utilizar para la solución. También ha sido necesario estudiar el código de la versión previa del simulador que iba a servir como base para este proyecto. Se incluyen además todas las tareas de planificación de trabajo y definición de objetivos a corto plazo debido a que era necesario realizar algunas pruebas iniciales antes de delimitar el alcance global del proyecto.
- **Fase de preparación:** en este período, una vez conocidos ya los requisitos necesarios para comenzar el desarrollo, se llevaron a cabo las tareas necesarias para instalar el entorno de desarrollo Unity3D e integrar en él, el dispositivo ASUS XtionPRO LIVE que será clave en el desarrollo del nuevo simulador. En esta etapa fue necesario realizar una

fase de investigación sobre diferentes plugins que permitieran realizar desarrollos en el entorno de trabajo Unity3D en el que se va a realizar la extensión funcional del simulador, puesto que el fabricante del dispositivo (ASUS) no los proporciona.

- **Fase de desarrollo:** durante esta fase se trabajó en el diseño y se desarrolló el código necesario para la implementación del entorno de simulación a partir de los requisitos del proyecto, permitiendo la comunicación entre usuario y avatar, mejorando considerablemente la sensación de inmersión. Además se comprobó el correcto funcionamiento de la plataforma 3DOF, los dispositivos HMD (Head-Mounted Displays) y el Joystick, dispositivos ya adaptados en la solución proporcionada, y se corrigieron aquellos detalles que hayan podido verse afectados al ampliar la funcionalidad del simulador.
- **Fase de finalización:** período que concluye el proyecto y que se compone de las tareas de control necesarias para verificar el cumplimiento de los objetivos planteados en la fase inicio. Se llevan a cabo baterías de pruebas exhaustivas de las funcionalidades del simulador y se analizan los resultados obtenidos. Una vez todo está validado, se procederá a dejar el simulador preparado para pruebas con usuarios reales y a elaborar la memoria.

A continuación se muestra un gráfico representativo del ciclo de vida del proyecto en relación al nivel de esfuerzo y el tiempo empleado en cada fase del mismo.

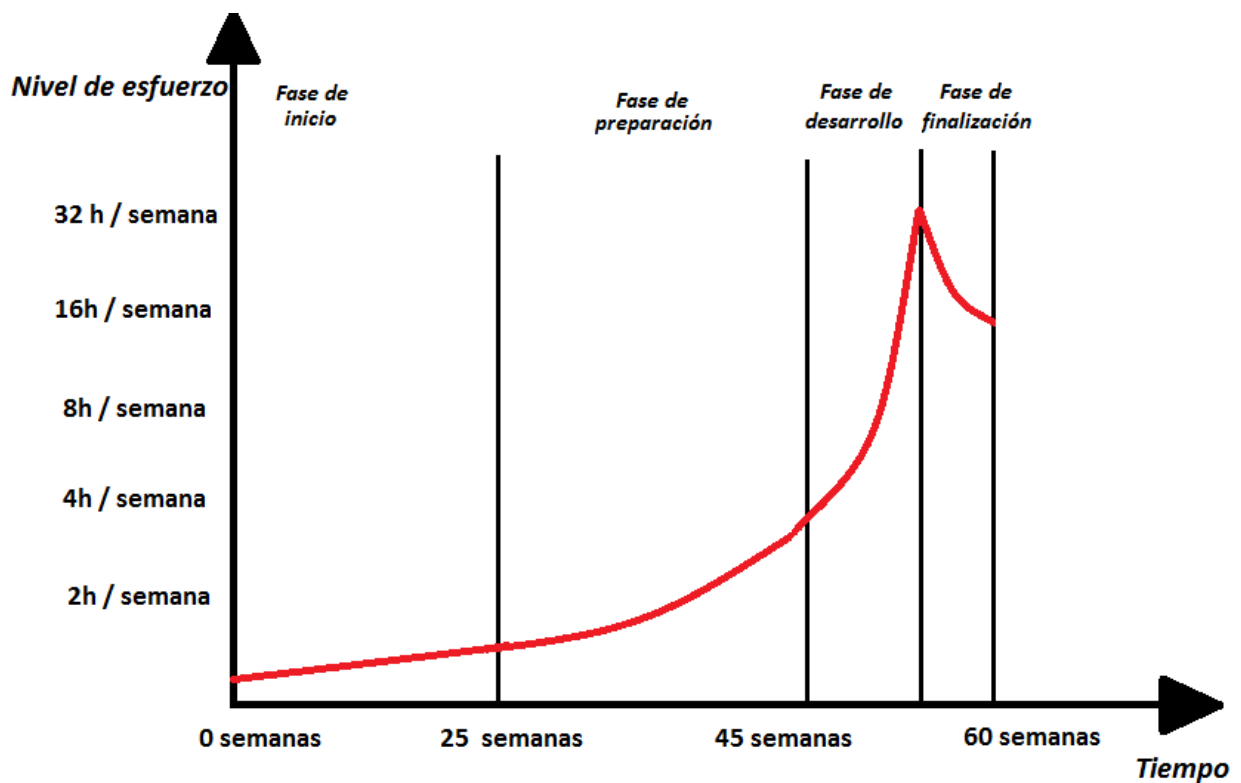


Figura 50: Ciclo de vida

5.2. Organización

Cada una de las cuatro etapas del proyecto (inicio, preparación, desarrollo y finalización) se subdivide a su vez en diferentes tareas, siguiendo un orden lógico para el correcto cumplimiento de objetivos.

La primera etapa estará relacionada con la familiarización del proyecto base y de las herramientas de desarrollo necesarias.

- Análisis del código proporcionado.
- Instalación del entorno de desarrollo Unity3D y lectura del manual de usuario en <http://docs.unity3d.com/Documentation/Manual/index.html>.
- Aprendizaje de la estructura básica de programación en C# (lenguaje de programación).

La segunda etapa se centrará en la integración del dispositivo de detección de movimiento y se compone de los siguientes hitos:

- Instalación de los drivers controladores necesarios para la detección como dispositivo externo de la ASUS XtionPRO LIVE.
- Programación necesaria para compatibilidad entre Unity3D y el dispositivo de detección de movimiento con la ayuda del plugin OpenSource proporcionado por Zigfu.

En la tercera etapa se han realizado las tareas necesarias para añadir la modalidad de entrenamientos:

- Importación de modelos 3D referentes a interiores de hogares, apoyándonos en la herramienta Sketchup de google, y edición posterior según sea necesario.
- Desarrollo del código para que el usuario vea reflejado sus movimientos en el personaje de simulación de la escena.
- Independización de los escenarios y físicas de los personajes
- Modificación del menú para añadir la opción de entrenamientos e incorporación de un gestor de cámaras.
- Incorporación de un sistema de ayudas y puntos de control.

En la cuarta etapa se ha procedido a realizar las pruebas necesarias que permitieran validar el proyecto:

- Pruebas, validación y análisis de resultados con usuarios reales.

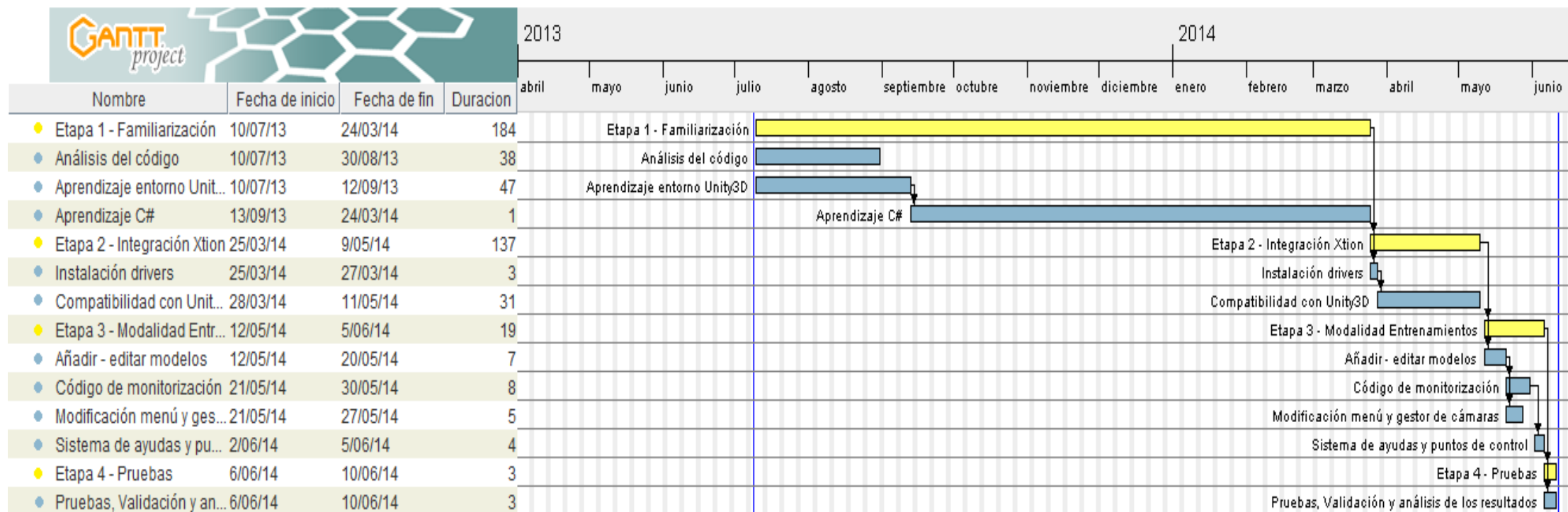


Figura 51: Diagrama de Gantt del proyecto

5.3. Plan de trabajo

Durante el desarrollo de este proyecto se ha necesitado del trabajo coordinado entre el desarrollador de la primera versión del simulador, el autor de este trabajo y la tutora de ambos. El tiempo estimado que se ha dedicado para la realización del trabajo es de unas 230 horas por parte del alumno y 60 horas por parte de la supervisora.

Este proyecto se inició con una reunión de ambos desarrolladores con la tutora, para conocer el simulador de partida y sus principales características y proponer la nueva línea de trabajo, que es la de posibilitar la monitorización del usuario.

Una vez presentada la idea base del trabajo y proporcionado el código base se estableció un período para familiarizarse con el código de partida, los recursos hardware a utilizar, el entorno de trabajo Unity3D y los lenguajes de programación necesarios (C# y JavaScript). Esta etapa no dispone de un tiempo específico debido a la fecha prematura en la que se produce (a principio del curso 2013-2014). Analizadas las opciones del nuevo hardware, se establecieron las posibles aplicaciones del nuevo simulador, todas ellas orientadas al campo de la medicina.

Una vez superadas las barreras propias de la primera toma de contacto con las herramientas a utilizar, se estableció un plan de trabajo con reuniones periódicas aproximadamente cada 3 semanas, que se compaginaron con un período de prácticas externas a la Universidad en la empresa de simulación SIMUMAK, responsable de la plataforma SMK3DOF [23]. El desarrollo conjunto de estas dos líneas de trabajo permitió una realimentación en ambos frentes, favoreciendo el surgimiento de nuevas metas. En principio estas reuniones se realizaron tanto personal como virtualmente.

A falta de unos 3 meses para la finalización del segundo cuatrimestre del curso y ya terminadas las prácticas, el plan de trabajo se intensificó y se asignó al alumno un puesto en un laboratorio con todos los recursos necesarios para la integración hardware de los diferentes elementos. Durante este período el

trabajo requirió de una comunicación más frecuente tanto con la tutora como con el desarrollador del trabajo.

El tiempo de trabajo aproximado fue de 1 hora al día en la etapa inicial (trabajando un día a la semana en el desarrollo, de forma más autónoma) y de 3 horas al día (trabajando tres días a la semana en la fase de desarrollo). La necesidad de compaginar el curso académico con la ejecución del proyecto provocó que el grueso del desarrollo se concentrara durante los últimos meses, período en el cual concluyeron las prácticas.

5.4. Presupuesto

En este apartado se va a realizar un desglose de los costes que has sido necesario afrontar para la consecución de la ampliación del simulador. Los costes se pueden organizar en función de su origen en personales, de equipo, de licencias y generales.

5.4.1. Coste de personal

Como se ha indicado en el apartado 5.3. Plan de trabajo, el desarrollo ha sido realizado gracias al trabajo conjunto de tres personas: la tutora (doctora), el desarrollador previo (titulado) y el desarrollador de prototipo actual (titulado). Utilizando el salario medio de remuneración que obtendrían y las horas de trabajo aportadas por cada uno, se puede deducir en la siguiente tabla (tabla 14) el coste total de los recursos humanos.

Tabla 17: Coste de personal

<i>Apellidos y nombre</i>	<i>Categoría</i>	<i>Dedicación (horas)</i>	<i>Coste hombre horas</i>	<i>Coste (Euro)</i>
Fernández Panadero, M ^a Carmen	Jefe de proyecto	60	40	2.400 €
de la Cruz Barqueroa, Valentín	Titulado	5	20	100 €
Díaz Jaime, José Antonio	Alumno	350*	20	7000 €
TOTAL		415	--	9500 €

*Además del desarrollo del simulador (230 horas), el alumno ha realizado esta memoria, que se estima en unas 120 horas de trabajo.

5.4.2. Coste del equipo

En este apartado se indican los costes asociados a todos los dispositivos hardware empleados en el desarrollo del simulador, tanto los que se han integrado como los necesarios para llevar a cabo el proyecto.

Tabla 18: Coste del equipo

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Período de depreciación	Coste imputable
Equipo de trabajo (Portátil)	600 €	100 %	10	60 meses	100 €
Asus XtionPRO LIVE	160 €	100 %	4	60 meses	11 €
Oculus Rift	230 €	50 %	1	60 meses	2 €
TOTAL					113 €

$$\text{Coste imputable} = \frac{\text{Nº de meses de dedicación}}{\text{Período de depreciación}} \times \text{Coste} \times \text{uso (\%)}$$

Se puede observar que los dispositivos con los que se realizó el desarrollo del simulador inicial (Vuzix, SMK3DOF y Joystick) no se tienen en cuenta en esta ampliación.

5.4.3. Coste de licencias

Cada uno de los programas presentados en el apartado 2.4. Tecnologías empleadas, conlleva un coste correspondiente a su licencia de trabajo. En la siguiente tabla (tabla 16) se desglosa el precio por licencia.

Tabla 19: Coste de licencias

PROGRAMA	PRECIO DE LICENCIA
Unity3D Pro	1.100 €
SketchUp	Gratuita
Blender	Gratuita
TOTAL	1100 €

5.4.4. Resumen de costes

Para finalizar y poder concluir con un coste total del proyecto, se va a realizar una suma de todos los costes indicados en los apartados anteriores y los impuestos asociados a los mismos. Además, se indica el coste general con y sin I.V.A.

Tabla 20: Coste general

Descripción	Presupuesto Coste Total
Coste de personal	9500 €
Coste imputable del equipo	113 €
Coste de licencias	1100 €

Costes indirectos (20%)	2143 €
SUBTOTAL (sin I.V.A.)	12856 €
I.V.A. (21%)	2700 €
TOTAL (con I.V.A.)	15556 €

Capítulo 6

Resultados y Conclusiones

El resultado que se ha obtenido una vez se ha completado el desarrollo de esta ampliación, es un simulador mucho más completo del que se partía y que permite aumentar la sensación de realidad percibida por el usuario, principal objetivo de los simuladores.

La monitorización permite reflejar en el entorno virtual desarrollado el movimiento real de quién lo utiliza. Esto unido al control a través de un Joystick y al sistema de gestión de cámaras (sobre todo la cámara de primera persona con posibilidad de giro) permite al usuario realizar ejercicios de la forma más cercana a la realidad posible (dentro de las limitaciones propias de estas tecnologías).

El simulador permite la realización de tres ejercicios de una dificultad variada, en los cuales se podrá realizar ejercicios orientados a la mejora de la motricidad de las extremidades superiores, y otros con el objetivo de facilitar el aprender las técnicas de realización de actividades cotidianas en entorno interiores, con la dificultad que conlleva el movimiento en estos recintos.

Además el sistema de puntos de control y de audio-ayuda actúa en todo momento de guía, otorgando “*feedback*” de los avances realizados e indicando el orden de los pasos a seguir para superar los ejercicios propuestos.

Integración ASUS XtionPRO LIVE

El primer paso por tanto que se debía realizar y que no iba a permitir iniciar el resto de vías de desarrollo hasta cumplimentar todos los objetivos, era el de integrar la ASUS XtionPRO LIVE en el entorno de desarrollo de Unity3D. Dado que el fabricante del dispositivo no facilitaba ningún plugin externo que lo permitiera, fue necesario investigar otras posibilidades, optando finalmente por utilizar el plugin de Zigfu para satisfacer este objetivo. Se buscaron varias fuentes que permitieran la comunicación Xtion-Unity3D, pero se encontró en Zigfu una herramienta para desarrolladores de carácter OpenSource que concedió las funcionalidades que se necesitaban para continuar con el desarrollo de la monitorización como tal. El principal inconveniente que presenta esta solución es una GUI (Graphical User Interface) que aparece por defecto al ejecutar al script principal de Zigfu (Zig.cs).

Desarrollo y modelado en Unity3D

Una vez conseguido el objetivo esencial, se continuó ampliando el escenario en el que se realizarían los ejercicios. Se importaron modelos y se modificaron de manera que se asemejaran a entornos interiores lo más realistas posibles. Además se diseñaron e implementaron todos los scripts necesarios para cumplir los requisitos expuestos en el apartado 3.6.1. Requisitos Funcionales. Se buscó que la ejecución real de los ejercicios se viera reflejada lo más fielmente posible en el entorno virtual y que los ejercicios supusieran un beneficio para el que los realizara. El grado de satisfacción alcanzado por quién lo probó y el hecho de que una vez introducido en la mecánica de realización de entrenamientos, los usuarios pidieran más posibles ejercicios, refleja que se ha cumplido este objetivo. El problema se ha resuelto desde el punto de vista técnico pero para obtener una solución más completa sería necesario ampliar el número de ejercicios y contar con la ayuda de profesionales de la salud para su diseño y supervisión.

Integración Hardware

Con el proyecto base desarrollado, aparecen objetivos de carácter “secundario” pero que suponen un gran avance para incrementar la sensación de inmersión en el entorno simulado. La integración de los dispositivos HMD (Head-Mounted Displays), tanto las Vuzix como las Oculus, convierten esta idea en realidad, sin la necesidad de llevar a cabo un trabajo demasiado costoso, ya que contaban con plugins compatibles con el desarrollo en Unity3D. Sin embargo la incomodidad, en el caso de las Vuzix, y la sensación de mareo en el caso de las Oculus Rift, impiden que estos dispositivos supongan una mejora aplicable a la ejecución de ejercicios durante un tiempo prolongado. Por todo ello, a veces una solución de baja tecnología como por ejemplo un simple monitor puede permitir un uso más prolongado del simulador en una misma sesión de entrenamiento.

Código simple y estructurado

Tratando de simplificar los scripts básicos requeridos para la monitorización, se buscó realizar estructuras modulares que permitieran su fácil compresión y posterior reutilización. Sin embargo, aquí se tuvo que llegar a un equilibrio entre:

- a) Generalizar lo más posible el sistema de control de ejercicios.
- b) Agilizar lo más posible la ejecución del simulador.

Finalmente se optó por priorizar la sincronización entre los movimientos del usuario y del avatar para estar lo más cerca posible del tiempo real de interacción, y a la vez tratar de modularizar los scripts lo máximo posible sin ralentizar el proceso de sincronización.

Objetivos generales – familiarización

Todo el desarrollo ha sido posible gracias a la familiarización con las tecnologías empleadas y los lenguajes de programación utilizados. Inicialmente no se tenía conocimientos de estas herramientas, y además de permitir obtener unos resultados exitosos, han permitido enriquecer las habilidades técnicas y de capacidad de abstracción del autor de la ampliación. Sin embargo, pese a que se han adquiridos conocimientos en nuevas tecnologías, estos han sido de carácter básico en comparación con las posibilidades que ofrecen programas como Blender y Unity3D.

Para finalizar, se puede concluir que tanto los objetivos funcionales como no funcionales se han conseguido completar de forma satisfactoria. Además una vez inmersos en el desarrollo han ido surgiendo nuevas metas hacia las que orientar el simulador y objetivos que mejoren el grado de realidad percibida que se exponen en el siguiente apartado (7. Futuras Líneas de Trabajo).

A nivel personal se ha conseguido satisfacer una serie de expectativas previas al desarrollo de este trabajo:

- Desarrollo de un proyecto complejo de inicio a fin.
- Ser capaz de abstraer la información esencial del trabajo de partida y ser capaz de integrar una ampliación nueva desarrollada.
- Llevar a cabo unas prácticas en una empresa.
- Aprender las diferentes formas de trabajo del resto de componentes del equipo y crear un grupo cohesionado y con ideas complementarias.
- Desarrollar una solución con posibles aplicaciones reales

Capítulo 7

Futuras Líneas de Trabajo

Durante el desarrollo del proyecto han ido surgiendo posibles mejoras y vías alternativas hacia las que orientar el trabajo. A continuación se hará una breve presentación de las mismas.

Mejora del movimiento de la silla de ruedas

Durante el proyecto el movimiento del usuario en la silla de ruedas no refleja el movimiento real, ya que su conducción está implementada de forma que la silla se mueve como un todo. Para intentar asemejarlo lo más posible a la realidad, en el caso de la modalidad de entrenamientos se ha ampliado el ángulo de giro controlado por el Joystick.

Sin embargo, el movimiento real de una silla de ruedas permite avanzar con cada una de las ruedas de manera independiente de modo que la rueda contraria pueda actuar de pivote, otorgando un ángulo de giro muy grande y favoreciendo la maniobrabilidad en las zonas interiores.

Por ello se propone diseñar un fichero de control extensible a cada una de las ruedas de forma que con la parte izquierda del Joystick se controle el movimiento de la rueda izquierda y con la parte derecha el de la rueda derecha.

Registro de entrenamientos

Esta funcionalidad extra consistiría en almacenar el movimiento del usuario durante la ejecución del ejercicio de entrenamiento, de forma que se guardara en un registro accesible a través del propio simulador de manera individualizada (utilizando nombre de usuario y contraseña).

Con esta extensión el simulador se podría utilizar como sistema de rehabilitación semi-profesional, ya que el usuario sería consciente de los avances que está realizando, al poder reproducir su ejecución de los ejercicios como si de un video se tratara. Esta nueva funcionalidad permitiría ejecutar en modo guiado hasta un determinado punto, por ejemplo donde el usuario se quede atascado, y pasar luego a modo libre para que el mismo usuario, o su entrenador, pudieran retomar el ejercicio desde allí.

Visibilidad monitorizada

Durante la memoria se citó que durante el desarrollo se pensó la posibilidad de anclar la cámara de primera persona a la cabeza del personaje virtual de forma que el movimiento real del cuello del usuario reflejara un cambio en la orientación de la cámara. Los problemas de vibración propios de la monitorización terminaron por provocar un cambio de esta funcionalidad y permitirla en giros controlados a través de pulsación de botones (mucho menos realista).

No se ha hecho especial hincapié en esto ya que su ausencia no supone un aspecto clave en el objetivo principal del desarrollo del simulador, sin embargo, en el caso de desarrollarse, sí supondría un gran avance en el posible grado de satisfacción del usuario y más aún si cabe si se compagina con los dispositivos HMD integrados.

Tratamiento de información

La principal diferencia entre el simulador desarrollado y los que se están comercializando actualmente en este mismo campo (monitorización con fines médicos, especialmente rehabilitación de pacientes discapacitados o con problemas de motricidad) es el tratamiento de datos objetivos como la aceleración, ángulo de movimiento y fuerza ejercida.

Esto permitiría a aquella persona que trate a un discapacitado, disponer de datos objetivos de las condiciones físicas iniciales del paciente y de las mejoras debidas al uso del simulador, pero esto requiere de la realización de un ensayo clínico que escapa al alcance de este proyecto.

Sin duda, el desarrollo de esta mejora supondría un giro brusco gracias al cual, el simulador pasaría de estar orientado a la familiarización a tener fines médicos reales y posibilitar su comercialización como posible producto complementario a la rehabilitación en clínicas privadas y hospitales.

Pruebas con discapacitados

Se tiene planificado una experiencia estructurada con usuarios discapacitados que seguramente supondrá una realimentación que permitirá corregir detalles y mejorar los ejercicios para lograr satisfacer en mayor medida el grado de satisfacción del usuario. Sin embargo esta tarea escapa al alcance de este proyecto.

Glosario

- **3DOF:** 3 Degrees Of Freedom
- **HMD:** Head Mounted Device
- **GPU:** Graphics Processing Unit
- **Háptico:** relativo al sentido del tacto
- **Prefab:** recurso que permite definir las propiedades de un objeto e instanciarlo múltiples veces
- **BIM:** Building Information Modeling
- **Respawn:** funcionalidad que permite, cuando el usuario se queda atascado en la escena, volver a una posición segura cercana al punto donde se produjo el atasco.
- **GUI:** Graphical User Interface

Bibliografía

- [1]Choi, B., & Baek, Y. (2011). Exploring factors of media characteristic influencing flow in learning through virtual worlds. *Computers & Education*,57(4), 2382-2394.
- [2] de la Cruz Barqueroa, V. (2013). TFG: Integración de la plataforma de movimiento SMK3DOF con UNITY 3D.
- [3] Fernández-Panadero, C., de la Cruz Barquero, V., Núñez, D. M., & Kloosa, C. D. (2013). PhyMEL-WS Wheelchair Simulator: A Preliminary Study to Increase Awareness about the Problems of Living the City in a Wheelchair.*London, 28-29 November 2013 King's College London, UK*, 210.
- [4] Csikszentmihalyi, M. (1988). The flow experience and its significance for human psychology.
- [5] Web del dispositivo de detección de movimiento utilizado en el proyecto: http://www.asus.com/es/Multimedia/Xtion_PRO_LIVE/ (20/05/2014)
- [6] Shannon, Robert; Johannes, James D. (1976). «Systems simulation: the art and science». *IEEE Transactions on Systems, Man and Cybernetics* 6(10). pp. 723-724.
- [7] Gálvez Mozo, A. (2004) Posicionamientos y puestas en pantalla. Un análisis de la producción de sociabilidad en los entornos virtuales. Barcelona: UAB.
- [8] Turkle, S. (1997) La vida en la pantalla. La construcción de la identidad en la era de internet. Barcelona: Paidós.
- [9] Web del simulador toyra: <http://www.toyra.org/que-toyra/> (22/05/2014)

[10] Web del subsistema asistido de toyra: <http://www.toyra.org/que-toyra/toyra-asistido/> (22/05/2014)

[11] Web del subsistema autónomo de toyra: <http://www.toyra.org/que-toyra/toyra-autonomo/> (22/05/2014)

[12] Ángel Gil-Agudo, Alberto Bernal-Sahún, Benito Peñasco-Martín, Beatriz Pérez-Aguilar, Ana de los Reyes-Guzmán, M^a del Carmen Herrero-Moreno, Ana Isabel de la Peña-González. “Sistema de realidad virtual aplicado a la rehabilitación de déficit motor en miembros superiores (TOYRA)”. 48º Congreso de la Sociedad Española de Rehabilitación y Medicina Física (SERMEF) celebrado en Las Palmas de Gran Canaria del 15 al 18 de Junio de 2010.

[13] Peñasco Martín B, de los Reyes Guzmán A, Gil Agudo A, Bernal Sahún A, Pérez Aguilar B., de la Peña González AI. “TOYRA: Realidad virtual para la rehabilitación de patologías de miembro superior”. XXVIII Congreso Anual de la Sociedad Española de Ingeniería Biomédica (CASEIB 2010), 24, 25 y 26 de noviembre de 2010, Madrid.

[14] Goodenough, R. R., Brooks, J. O., Pagano, C., & Evans, D. F. (2012). Display configurations in a small footprint clinical driving simulator. *Displays*, 33(4), 196-205.

[15] Artículo del simulador del CIDIF: <http://www.abc.es/local-castilla-leon/20140206/abci-aspaym-desarrolla-simulador-virtual-201402052118.html> (22/05/2014)

[16] Artículo del simulador de la PUCP: <http://puntoedu.pucp.edu.pe/noticias/invento-desarrollado-en-la-pucp-para-el-tratamiento-de-personas-con-discapacidades-motrices/> (22/05/2014)

[17] Web de consulta: [http://es.wikipedia.org/wiki/Unity_\(software\)](http://es.wikipedia.org/wiki/Unity_(software)) (24/05/2014)

[18] Web de consulta: <http://es.wikipedia.org/wiki/SketchUp> (24/05/2014)

[19] Web de consulta: <http://es.wikipedia.org/wiki/Blender> (24/05/2014)

[20] Web de consulta: <http://www.fnac.es/Asus-Xtion-PRO-LIVE-Accesorios-informatica-Accesorios-Micro/a731095> (20/05/2014)

[21] Google SketchUp Gallery

- [22] Web oficial de Unity3D (apartado de requisitos del sistema)
<http://unity3d.com/es/unity/system-requirements> (24/05/2014)
- [23] Web del proveedor de la plataforma SMK3DOF: <http://simumak.com/es/>
(24/05/2014)
- [24] Web de la Organización Mundial de la Salud
http://whqlibdoc.who.int/publications/2012/9789243548050_Introducci%C3%B3n_spa.pdf?ua=1 (28/05/2014)
- [25] Abellard, P. et al. Electric Wheelchair Navigation: why?, when?, how? In: MILELLA, ; DI PAOLA, D.; CICIPELLI, Mechatronic Systems Applications. 2010. Cap. 5. ISBN: 978-953-307-040-7.
- [26] European Commission. Horizon 2020. Tackling Societal Challenges:
<http://ec.europa.eu/research/horizon2020/index_en.cfm?pg=better-society>.
Visited: Aug. 2013.
- [27] Pivik, J. et al. Using Virtual Reality to Teach Disability Awareness. Journal of Educational Computing Research, v. 26, n. 2, p. 203-218, 2002. ISSN DOI: 10.2190/WACX-1VR9-HCMJ-RTKB
- [28] Web de la organización Mundial de la Salud (paquete básico de entrenamiento):
<http://www.who.int/disabilities/technology/wheelchairpackage/en/> (28/05/2014)
- [29] Web de la organización Mundial de la Salud (paquete intermedio de entrenamiento):
<http://www.who.int/disabilities/technology/wheelchairpackage/wstpintermediate/en/> (28/05/2014)
- [30] Web de la Universidad Carlos III de Madrid (apartado de protección de datos)
http://portal.uc3m.es/portal/page/portal/inicio/universidad/proteccion_datos
(28/05/2014)
- [31] Real Decreto 1276/2011, de 16 de septiembre, de adaptación a la normativa de Convención Internacional sobre los derechos de las personas con discapacidad: <http://www.boe.es/boe/dias/2011/09/17/pdfs/BOE-A-2011-14812.pdf>
(28/05/2014)

[32] Rerc's Advisory Board. The WC19 Information Resource: Crash-tested Wheelchairs & Seating Systems. <<http://www.rercwts.pitt.edu/WC19.html>>

Anexos

Anexo A: Manual de usuario

Este manual complementa el manual de la versión inicial del simulador [2].

- **Menú Inicial:**

1) Selección de modo de juego: se ofrecen tres posibles modalidades a elegir. Para desplazarte entre las opciones se puede usar las flechas de dirección y/o el mando izquierdo del Joystick hacia arriba o abajo y pulsar “Enter” o el botón 2 para confirmar.



Figura 52: Selección de modo de juego (Menú Inicial)

2) Selección de género: permite elegir el sexo del personaje a utilizar. El mismo desplazamiento y selección que en la selección de modo de juego.



Figura 53: Selección de género (Menú Inicial)

3) Selección del momento del día: permite elegir el horario en el que se realizará la simulación, día o noche. La selección sigue el mismo modo de desplazamiento y selección que en la selección de género.



Figura 54: Selección del momento del día (Menú Inicial)

- **Menú Pausa (Disponible para modo guiado y libre)**

El control del desplazamiento es el mismo que el utilizado en el Menú Inicial. Para confirmar pulsar “Enter” o el botón 2 y para seleccionar las flechas de dirección o el mando izquierdo del Joystick.



Figura 55: Menú Pausa

1) **Continuar:** permite continuar con la ejecución del ejercicio

2) **Logros:** muestra los retos superados hasta el momento.



Figura 56: Registro de logros (Menú Pausa)

3) **Selección de escena:** vuelve al Menú Inicial.

4) **Salir:** termina con la ejecución del simulador.

- **Respawn:**

Las tres modalidades de las que dispone el simulador, permiten en el caso de que el usuario volcara o se quedara atrapado en un determinado obstáculo, reiniciar su posición de partida en el caso de los ejercicios de entrenamiento²³, o volver al punto de control más cercano en el caso del modo libre pulsando el botón 1 o la tecla “R”.

²³ En este caso sólo será posible hacer respawn siempre que le usuario no se coloque en una posición específica o punto de control que actúe de anclaje

- **Modalidad ENTRENAMIENTOS:**

1) Selección de ejercicio: permite elegir entre tres ejercicios diferentes. Para desplazarte entre las opciones se puede usar las flechas de dirección y/o el mando izquierdo del Joystick hacia la izquierda o la derecha.



Figura 57: Selección de ejercicio

2) Confirmación de ejercicio: una vez escogido el ejercicio a realizar, confirmar pulsando “Enter” o el botón número 2 del Joystick. En este momento se cargará el personaje en la escena y comenzará la monitorización. Cuando el usuario ya esté captado, pulsar “V” o el botón 4 para iniciar el ejercicio.

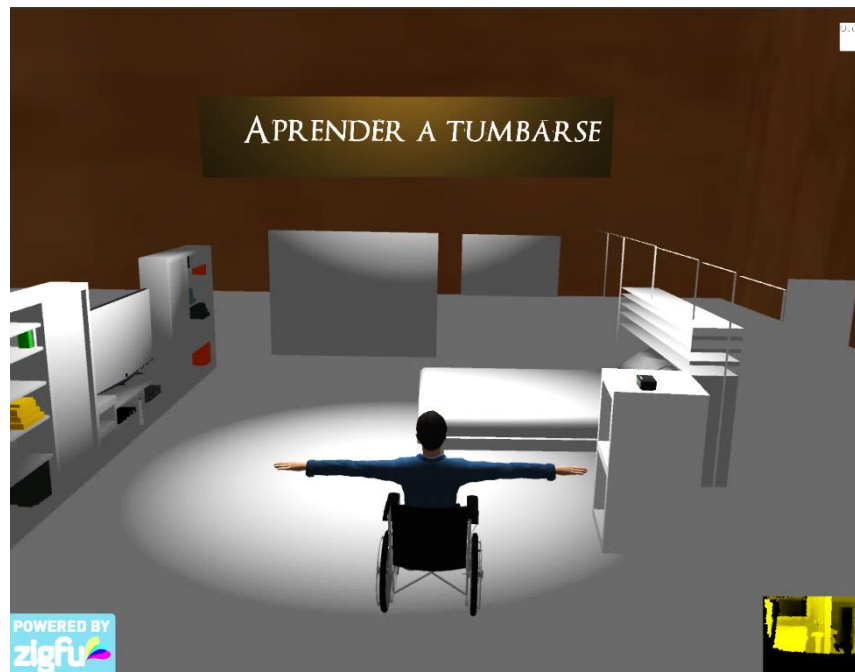


Figura 58: Ejercicio confirmado y comienza la monitorización

3) Navegación: para cambiar la cámara se pulsará la tecla “V” o el botón 4.

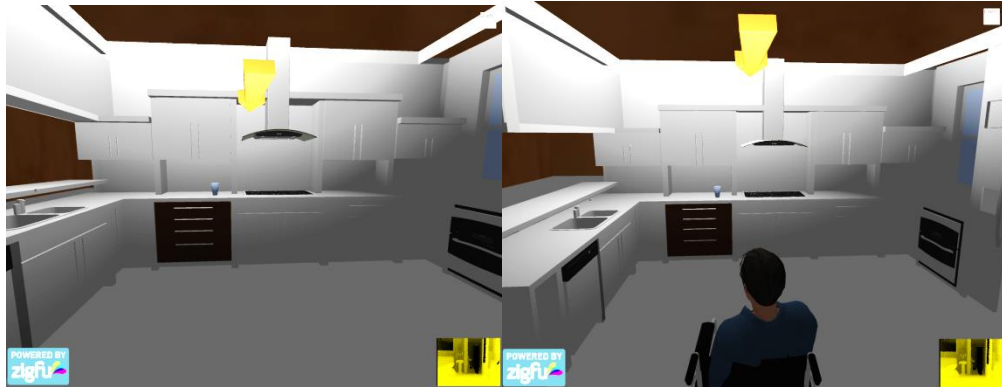


Figura 59: Cambio de cámara (Modalidad de entrenamientos)

En el caso de no tener las Oculus conectadas, en la cámara de primera persona se podrá girar la visión hacia la izquierda pulsando “O” o el botón 1 y hacia la derecha pulsando “P” o el botón 3.

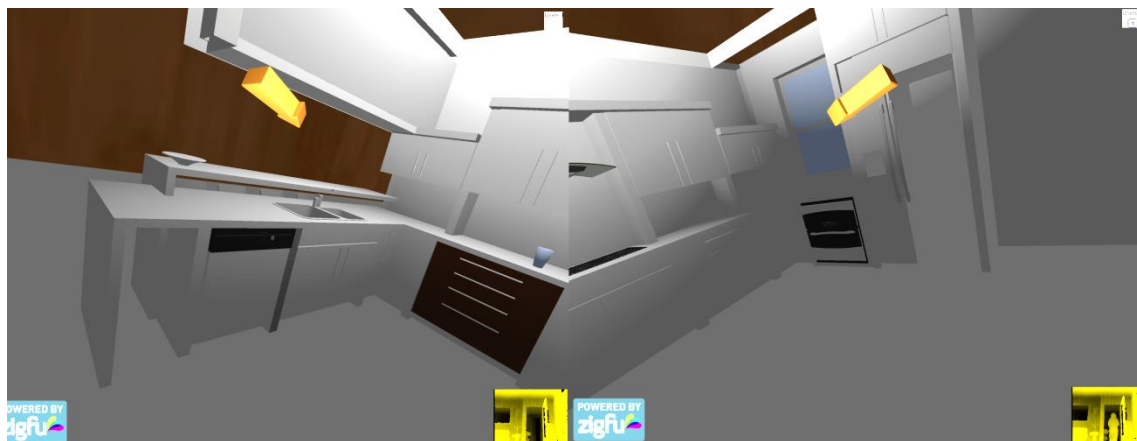


Figura 60: Giro de cámara (Modalidad de entrenamientos)

3) Pausa: si fuera necesario pausar el ejercicio, pulsando la tecla “M” o el botón 3 se detendría temporalmente, hasta que se volviera a pulsar otra vez la misma tecla o botón. Para que el usuario tenga consciencia de tener el modo

pausa activado, aparecerá un texto en la esquina superior izquierda que lo indicará.

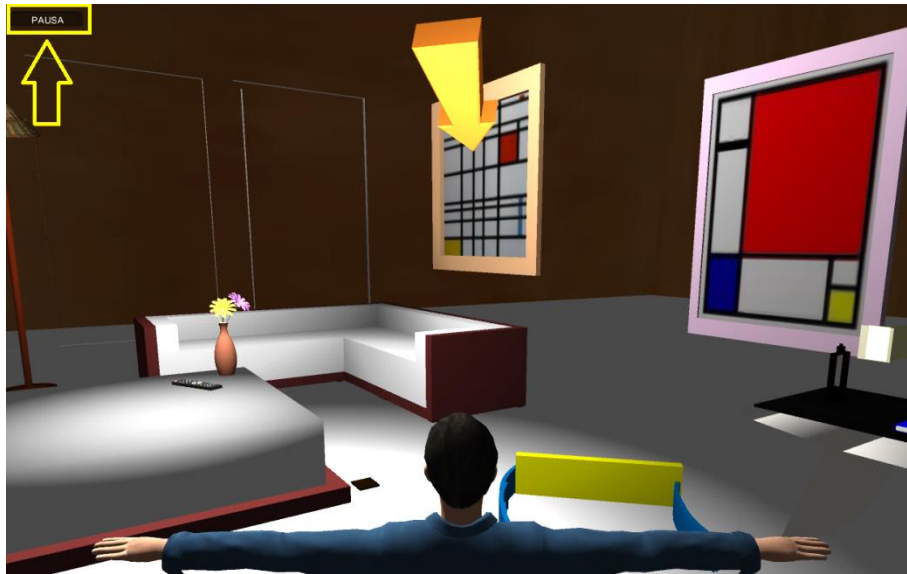


Figura 61: Ejercicio entrenamiento pausado

- **Controles:**

- 1) **Teclado:**

Flechas	→	Movimiento / Selección
Enter	→	Confirmar
M (*)	→	Menú Pausa
R	→	Respawn
V	→	Cambio de vista
O (**)	→	Giro de cámara (Dch.)
P (**)	→	Giro de cámara (Izq)

* En la modalidad de entrenamientos no se mostrará el menú, únicamente se parará la monitorización y el movimiento de la silla

** Sólo disponible en la modalidad de entrenamientos

2) Joystick:

Mando izq.	→	Movimiento / Selección
2	→	Confirmar
3	→	Menú Pausa
1	→	Respawn
4	→	Cambio de vista
1 (*)	→	Giro de cámara (Dch.)
3 (*)	→	Giro de cámara (Izq)

* Sólo disponible en la modalidad de entrenamientos



Figura 62: Controles del Joystick

Anexo B: Manual de desarrollador

- **Instalación Hardware:**

Plataforma 3DOF

Se comunica con el ordenador a través de un cable Ethernet que funciona mediante sockets y es necesario configurar la red LAN para que funcione correctamente.

- 1) Abrir el Panel de control.
- 2) Click en Redes e Internet y abra el Centro de redes y recursos compartidos.
- 3) Click en Cambiar la configuración del adaptador y se mostrarán las redes utilizadas por el ordenador.

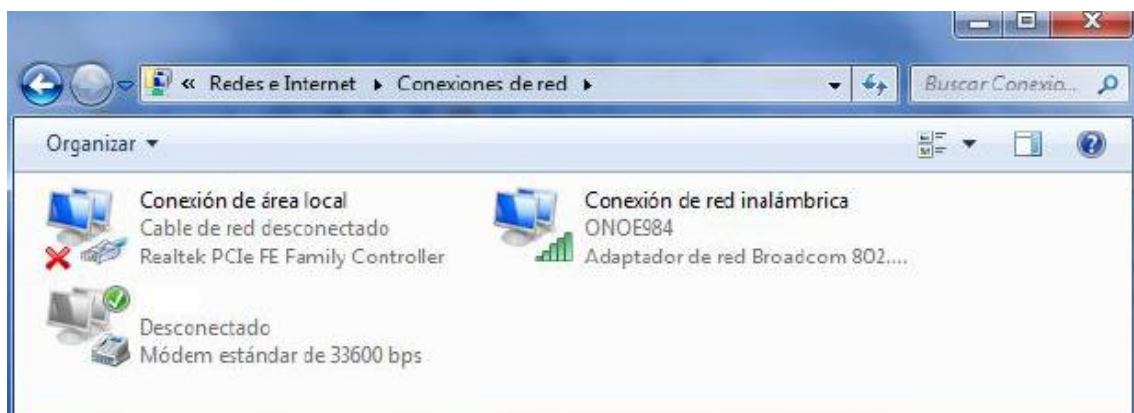


Figura 63: Conexiones de red

- 4) Click con botón derecho en conexión de área local y click en propiedades. Se mostrará la siguiente ventana:

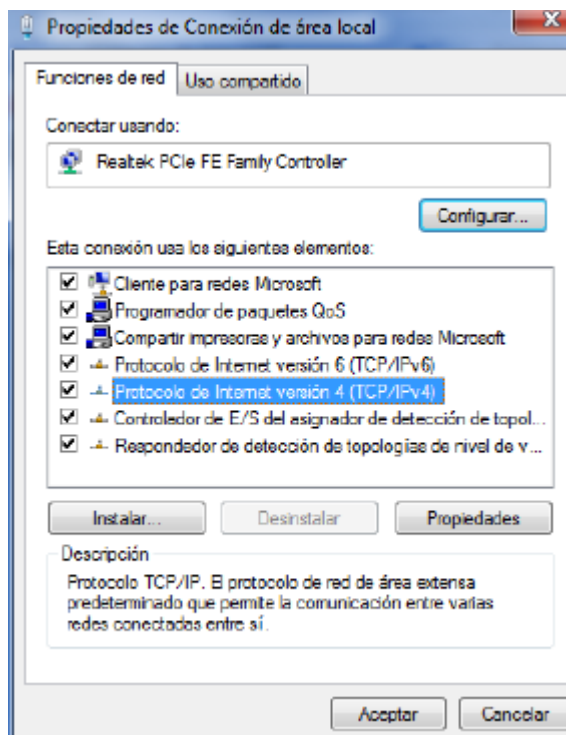


Figura 64: Propiedades de Conexión de área local

5) Seleccionar Protocolo de Internet versión 4 (TCP/IPV4) y click en propiedades. Se mostrará la siguiente ventana:

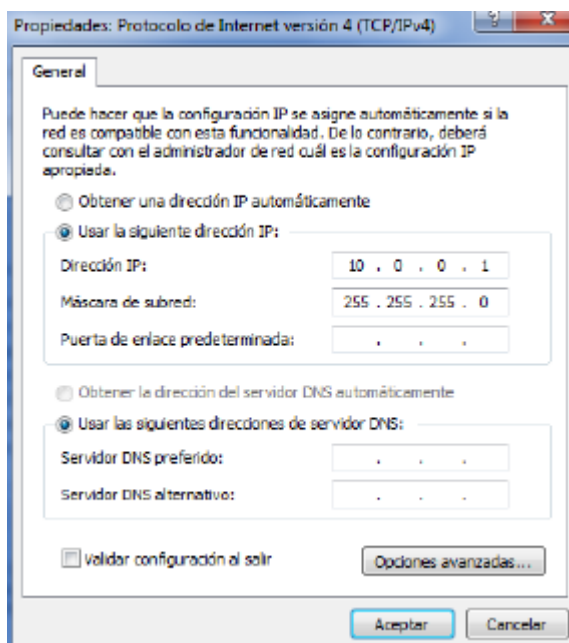


Figura 65: Propiedades de protocolo (TCP / IPV4)

6) Asignar la dirección IP de la plataforma (10.0.0.117) y la máscara de subred 255.255.255.0.

Si con esta configuración no logra conectarse con la plataforma, puede ser problema del Firewall de Windows o del antivirus empleado. Asegúrese de que en ambos casos están desactivados.

Aparte de la configuración, es necesario disponer de los archivos “SMK3DOF_HOSTDRIVER” (librería de comunicación) y “SMK3DOFConfig” (fichero de comunicación .ini) en el directorio raíz del proyecto durante el desarrollo y en el directorio contenedor de la *Build* durante la fase de despliegue. En el caso de necesitar configurar las propiedades de la plataforma (como por ejemplo la suavidad de movimiento, modos predeterminados de vibración, etc), es necesario incluir en dichos directorios el fichero de texto “parametros3DOF.txt”.

ASUS XtionPRO LIVE

Es necesario instalar los plugins necesarios citados en 3.5.1. Dispositivo de detección de movimiento. Para ello el dispositivo cuenta con un CD de instalación que se introducirá en el lector del ordenador y únicamente será necesario seguir los pasos de instalación. Una vez ya instalado, se podrá utilizar.

TrackIR 5

Es necesario descargar el software de la página oficial de Natural Point (<http://www.naturalpoint.com/trackir/06-support/support-download-software-and-manuals.html>) y realizar el proceso de instalación.

Según el dispositivo de rastreo utilizado se seguirán unos pasos u otros:

- **Track Clip:** el montaje de este reflector va solidario a una gorra.



Figura 66: Montaje gorra y reflector (Track Clip)

- **Track Clip Pro:** es necesario conectarlo al ordenador a través de un conector USB.

La cámara de TrackIr deberá fijarse dirigida de frente a la cabeza del usuario y cerciorarse de que se reciben los tres puntos verdes procedentes del reflector en el programa indicado anteriormente. Para actualizar la posición de la cabeza pulsar la tecla F12.

Vuzix VR920

Para su instalación únicamente es necesario conectar el cable VGA y USB al ordenador y automáticamente se instalará el controlador necesario.

Oculus Rift VR

Para su instalación es necesario introducir el CD de instalación proporcionado con el dispositivo e iniciar el proceso de instalación. Una vez completado, las gafas ya están disponibles para utilizarse.

Logitech Dual Action

Para su instalación únicamente es necesario conectar el cable USB al ordenador y automáticamente se instalará el controlador necesario.

- **Instalación Software:**

Para ejecutar el proyecto es necesario tener instalado el motor de desarrollo de Unity3D. Se recomienda utilizar una versión superior a la 4.0, ya que algunas funcionalidades (por ejemplo los controladores de las Oculus) son posteriores a esta versión.

Para descargarlo, se podrá hacer desde la página web oficial del programa en el enlace: <http://unity3d.com/unity/download>.

- **Estructuración del código:**

AdminEscena.cs

Script principal que controla la ejecución de las modalidades guiado y libre. Se basa en constantes para que el sistema sepa que opciones se han seleccionado.

```
public static int modoJuego=GUIADO;  
public static int InitJuego=MENU;  
public const int GUIADO=1;  
public const int LIBRE=2;  
public const int ENTRENOS=3;  
public const int MENU=0;  
public const int JUGANDO=1;
```

Figura 67: Constantes de control (AdminEscena.cs)

En él se gestionan la creación de instancias de los personajes a utilizar y el anclado a la silla de ruedas para disponer de un movimiento solidario.

```
//Destruimos el personaje (Clon) anterior en caso de que lo hubiera
if(MujerClon!=null)
    Destroy(MujerClon.gameObject);
if(HombreClon!=null)
    Destroy(HombreClon.gameObject);

//Creamos un personaje (Clon) mujer u hombre dependiendo de la seleccion inicial
//Esto nos permitira simplificar el posterior proceso de reseteo de la escena
if(MenuInicio.esMujer){
    MujerClon=Instantiate(Mujer,Vector3.zero,Quaternion.identity)as Transform;
    MujerClon.gameObject.SetActive (true);
}
else{
    HombreClon=Instantiate(Hombre,Vector3.zero,Quaternion.identity)as Transform;
    HombreClon.gameObject.SetActive (true);
}
```

Figura 68: Instancia de personajes (AdminEscena.cs)

Por último, su ejecución se base en el método Update() por defecto de C#, repetido una vez por cada frame procesado, en el cual se van llamando a métodos que gestionan todo el proceso:

- FijarPosicion() → Establece la posición inicial de la silla.
- Actualizar_posicion() → Fija el personaje a la silla.
- pulsarPausa() → Gestiona el pulsado de pausa.
- cambioCamara() → Gestiona el cambio de visión.
- respawn() → Gestiona elección de respawn.
- reset() → Reinicia el ejercicio.

```
// Update is called once per frame
void Update () {
    if(InitJuego == MENU){
        FijarPosicion();
    }
    if (InitJuego == JUGANDO){
        Actualizar_posicion();
    }
    pulsarPausa();
    cambioCamara();
    respawn();
    if(resetEnabled)
        reset(); //Reseteamos todos
    if(Input.GetKeyDown(KeyCode.Escape))
        Application.Quit(); //Cerramos
}
```

Figura 69: Update AdminEscena.cs

ManageExercises.cs

Script principal que controla la ejecución de la modalidad de entrenamientos y la gestión de sus ejercicios. Al igual que AdminEscena.cs, se basa en constantes que indican el estado de la ejecución.

```
public static int estado;
public const int SELECCION = 0;
public const int MENU = 1;
public const int JUGANDO = 2;
public const int TERMINADO = 3;
public const int ESPERA = 4;
```

Figura 70: Constantes de control (ManageExercises.cs)

Cada ejercicio cuenta con un muro, una cámara y una luz de presentación, pasados al script a través del inspector. Este fichero se encarga de gestionar su activación y desactivación en función de la selección del ejercicio.

```

void seleccion(){
    if((Input.GetKeyDown(KeyCode.RightArrow) && contRef>4)||((Input.GetAxisRaw("Horizontal")>0.9)&&contRef>5)){
        desactivar_ejercicio(ejercicio);
        if(ejercicio<2)ejercicio++;
        activar_ejercicio(ejercicio);
        audio.PlayOneShot(tick);
        contRef = 0;
    }
    if((Input.GetKeyDown(KeyCode.LeftArrow) && contRef>4)||((Input.GetAxisRaw("Horizontal")<-0.9)&&contRef>5)){
        desactivar_ejercicio(ejercicio);
        if(ejercicio>0)ejercicio--;
        activar_ejercicio(ejercicio);
        audio.PlayOneShot(tick);
        contRef = 0;
    }
    if(Input.GetKeyDown(KeyCode.Escape) && contRef > 15){
        contRef = 0;
        audio.PlayOneShot(tick);
        Application.Quit();
    }
    if(Input.GetKeyDown(KeyCode.Return)||Input.GetKeyDown(KeyCode.Joystick1Button1)){
        estado = ESPERA;
        AudioListener listener;
        listener = (AudioListener)recipiente.GetComponent("AudioListener");
        listener.enabled = false;
        listener = (AudioListener)Entrenamientos[ejercicio].GetComponent("AudioListener");
        listener.enabled = true;
    }
    contRef ++;
}

void activar_ejercicio (int pos){
    Muros[pos].SetActive(true);
    Camaras[pos].SetActive(true);
    Lights[pos].SetActive(true);
}

void desactivar_ejercicio(int pos){
    Muros[pos].SetActive(false);
    Camaras[pos].SetActive(false);
    Lights[pos].SetActive(false);
}

```

Figura 71: Selección de ejercicio (ManageExercises.cs)

Por último, su ejecución se base en el método Update() por defecto de C#, repetido una vez por cada frame procesado, en el cual se van llamando a métodos que gestionan todo el proceso haciendo uso de un switch.

- selección() → Para elegir ejercicio.
- comienzo() → Se sitúa la silla y el personaje en el ejercicio.
- colocar() → Actualiza la posición y rotación del personaje.
- cambiar_camara() → Gestiona el cambio de vista y rotación.

```
// Update is called once per frame
void Update () {
    switch (estado){
        case SELECCION:
            seleccion();
            break;
        case ESPERA:
            comienzo();
            break;
        case JUGANDO:
            colocar();
            cambiar_camara();
            if(fin_ejercicio == 2)estado = TERMINADO;
            break;
        case TERMINADO:
            #if UNITY_EDITOR
                UnityEditor.EditorApplication.isPlaying = false;
            #endif
            break;
    }
}
```

Figura 72: Update ManageExercises.cs

Scripts de ejercicios (modalidad de entrenamientos)

Basan su funcionamiento en métodos `OnTriggerEnter` dentro de los cuales se detecta si el elemento colisionado es el de interés a través del nombre.

```
// Cuando se produce una colision con un collider trigger
void OnTriggerEnter(Collider col){
```

Figura 73: Método OnTriggerEnter (Ejercicios monitorizados)

Gestionan la jerarquía de los objetos colisionados con la propiedad `“transform.parent”` y el color de las texturas de sus materiales con `“transform.renderer.material.setColor(“_Color”, Color.*)”`.

```
//Priemer paso del ejercicio
if(col.gameObject.name == "RightLeg" && elemento.name == "pie1" && onepick == 0){

    col.transform.parent      = elemento.transform;
    col.transform.localScale  = Vector3.one;
    PickFoot.pick            = 1;
    onepick                  = 1;
    elemento.transform.parent.renderer.material.SetColor("_Color",Color.green);
    CompassTrain1.stepsNumber++;

}
```

Figura 74: Gestión de jerarquías y colores (Ejercicios monitorizados)

Hacen uso de constante internas para controlar el estado del objeto de interés colisionado o no y la colocación de la silla de ruedas en la posición correcta.

```
//Variables para detectar colisiones
public GameObject elemento;
public static int onepick = -1;
public static bool colocado = false;
```

Figura 75: Constantes de control (Ejercicios monitorizados)



Universidad
Carlos III de Madrid

Department of Telematics Engineering

FINAL DEGREE PROJECT

Integration of motion sensor in Unity-3D to monitor the mobility of wheelchair users

Author: José Antonio Díaz Jaime

Tutor: M^a del Carmen Fernández Panadero

Leganés, June 2014

Index

Abstract	3
Introduction	4
Results and conclusions	6

Abstract

Technologies such as simulation, allow working in a secure environment that protects the user all time. Simulation has mainly risen in popularity in the environment of medical-therapy, where the opportunity to keep real-time tracking of patients or testing with low personal and economic costs, represent a great alternative to traditional methods.

In this project, it has been decided to apply simulation technologies to monitor the movement of a disabled user so that the feeling of immersion is as close as possible to reality. Development includes hardware integration tasks (which highlights a motion detection device) and software, using the Unity3D game engine as the main tool.

It has been possible to obtain a simulator that allows users to visualize their movement in a virtual environment while doing interactive and guided exercises in a scene according to the theme of the simulation. In this way, the project has become an effective solution for learning and familiarization with common activities in the disabled people.

Finally, this simulator allows the users to experience for themselves the difficulties that disabled people face when performing tasks that are simple for a person with full mobility.

Keywords: Simulator, disabled, monitoring, learning, medicine, therapy.

Introduction

The digital and boundless technological development era that we are currently living in, cause the emergence of new devices that satisfy needs that were previously unaware of such as the availability of internet in mobile phones, remotely controlling home appliances, screens with 3D vision, etc.

In this environment, one of the industries that is experiencing a development and commercial boome, has appeared: simulators. In Spain there are several major companies like Indra, traditionally dedicated to the world of simulation, but in recent decades other smaller companies as Simumak (Simumak) and Future Platone (FP) are bringing the world of simulators to customers with less commercial capacity, contributing to the expansion of this technology.

This tool provides many possibilities in the world of learning because it can engage the subject physically, mentally and emotionally. Proof of this is the huge number of applications for this technology in different areas such as medicine, sports, and recreation therapy.

Much of the potential of this technology is that it allows to recreate real situations in virtual environments providing safety to the user who controls it. It also allows users to improve their skills through practice, in order to later apply them to their daily lives. The impossibility of failure safely makes simulators an ideal learning tool by competencies. Although most cases require a real learning experience to complete the learning, the usefulness and advantages that simulators provide is obvious, particularly as an initial contact tool to accelerate learning in cases where real-world practice is not initially possible.

One of the main features of the simulators, representing a decisive factor to satisfy the needs of the user, is to achieve the highest level of realism possible, allowing the user to forget that a virtual tool is being used and making the experience as rewarding as possible. Authors like Choi [1] have demonstrated the impact of interactivity and realism in student satisfaction and therefore in learning outcomes.

However, leaving the benefits of interactivity aside, making use of simulators allows the user to perform competency training and, in the case of this simulator, instrumental skills. These skills play an fundamental role and may be methodological, physical or cognitive.

The simulator developed in this project allows the user to acquire a range of skills such as:

- Dealing with complex problems to contribute to the ability to make reasoned decisions and solve problems.
- Physical skills related to the use of the simulator itself.
- Training on simple tasks that allow analysis and assimilation capacity of the methodologies used in each exercise.

It can be ascertained that the tool that is going to be designed will enable disabled users to identify the skills needed to perform successfully in the real world, while users without disabilities will become aware of the difficulties presented by seemingly simple exercises.

Results and conclusions

The result once the development of this extension has been completed is a much more complete and competent simulator that increases the sense of reality perceived by the user, which is the main goal of simulators.

Monitoring allows us to reflect the real movement of the user in the virtual environment. This, coupled with the control via joystick and the camera management system (especially the first-person camera rotatably), allows users to perform exercises in a way that brings them closer to reality (within the limitations of these technologies).

The simulator allows the completion of three exercises of varied difficulty which can be aimed at improving upper limb motor abilities, as well as other exercises in order to learn techniques to allow performing daily activities in indoor environments with the difficulty of movement in these areas.

Furthermore, the system of checkpoints and audio-aid acts as a guide at all times, giving "feedback" of progress and indicating the order of the steps to overcome the exercises.

ASUS XtionPRO LIVE integration

The first step that needed to be done and that would not allow the other phases of the project to start being developed, was to integrate ASUS XtionPRO LIVE in the Unity3D development environment. Since the device itself did not provide any external plugin that allowed this, it was necessary to

investigate other possibilities, finally opting to use the Zigfu plugin to meet this objective. Several sources that allowed the Xtion-Unity3D communication were searched, but an OpenSouce tool for developers was found in Zigfu which granted the functionalities that were needed to continue with the development of the monitoring system itself. The main drawback of this solution is a GUI (Graphical User Interface) that appears by default when you run the main Zigfu script (Zig.cs).

Development and modeling in Unity3D

Once the essential goal was achieved, we continued expanding the scenario in which the exercises would take place. Models were imported and modified to be as realistic and close to indoor environments as possible. All the necessary scripts to meet the requirement were also designed and implemented . It was sought that the execution of the exercises reflected reality as closely as possible in the virtual environment and that the exercises were beneficial to the users. The degree of satisfaction achieved by those who tried it and the fact that once introduced into the mechanics of performing training exercises, subjects asked for more, shows that this objective has been fulfilled. The problem has been solved from a technical point of view, but to get a more complete solution it would be necessary to expand the number of exercises and enlist the help of health professionals for design and supervision.

Hardware integration

With the base project developed, secondary goals appear that imply great improvements towards increasing the sense of immersion in the simulated environment. The integration of HMD devices (Head-Mounted Displays), both Vuzix and Oculus, turn this idea into reality, without the need to engage in expensive work, because they had plugins compatible with development in Unity3D. However, the discomfort, in the case of Vuzix, and dizziness in the case of Oculus Rift keep these devices from becoming a solution to exercises

that need to be performed for long periods of time. Therefore sometimes low-tech solutions such as a single monitor can allow a longer use of the simulator in a single training session.

Simple and structured code

Trying to simplify the required basic scripts for monitoring was sought to perform modular structures that allow easy understanding and reuse. But in this case a balance had to be found between:

- a) Generalizing the control system of exercises as much as possible.
- b) Expediting the execution of the simulator as much as possible.

Finally, it was decided to prioritize the synchronization between the movements of the user and the avatar for them to be as close as possible to real time interaction, and at the same time to try to modularize scripts as much as possible without slowing the synchronization process down.

General objectives – familiarity

All the development has been made possible through familiarization with the technologies and programming languages that were used. Initially no knowledge of these tools was possessed, and in addition to allowing to obtain successful results during the project, learning them has enriched the technical capacities and abstraction abilities of the author of the extension. However, although a great deal of knowledge in new technologies has been acquired, this is basic in nature compared to the possibilities offered by programs like Blender and Unity3D.

Finally, we can conclude that the achievement of technical, functional and general objectives has been successfully completed. Moreover, once involved in development new goals have emerged to guide the simulator or

targets that improve the degree of perceived reality as outlined in the following section.

Personally, it has been possible to satisfy a number of expectations that the author had before the development of this project

- Development of a complex project from start to finish.
- Be able to abstract the essential starting job information and be able to integrate a new extension developed.
- Undertake an internship at a company.
- Learn different working methods from the other components of the team and create a cohesive group with complementary ideas.
- Develop a solution with potential real applications.